NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

63-3-4

Ford Motor Company
AERONUTRONIC DIVISION

Ford Motor Company
AERONUTRONIC DIVISION

Publication No. U-1909

ESD-TDR-62-340

SIMULATION ERROR ANALYSIS SYSTEM

PROGRAMMING DOCUMENT

Prepared for:

496-L System Program Office
Electronic Systems Division
Air Force Systems Command
United States Air Force
Lawrence G. Hanscom Field
Bedford, Massachusetts

Under Contract: AF 19(604)-7375

30 November 1962

FOREWORD

This document presents the theory, program philosophy, and operating instructions for the Simulation Error Analysis System (COSINE 5B). As a joint effort undertaken by the RCAF Directorate of Systems Evaluation and the USAF 496L Systems Project Office, the computer system, developed at Aeronutronic, provides an automatic means with which the observations of a specified orbit obtained from a sensor network may be simulated and analysed, yielding the uncertainty in the orbit elements due to the observational inaccuracies. The system also provides the capability of predicting future position and velocity errors caused by the uncertainty in the orbit elements.

The presentation of the document is made in loose leaf binder form with the understanding that the system will not remain static. Rather, it will be in a constant state of modification, adhering to requirements for specialized simulations and improved error analysis techniques. With this design concept therefore, the user may anticipate a series of revisions for future incorporation in the program document.

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

ILLUSTRATIONS (Continued)

SECTION 1

INTRODUCTION

A cooperative program of work has been undertaken by the RCAF Directorate of Systems Evaluation (DSE) and the USAF 496L Systems Project Office to provide a background of knowledge which can be applied to a broad range of problems concerning satellite detection, tracking, orbit determination, and prediction of future satellite positions. Aeronutronic has assumed a major portion of the program development through the 496L SPO, with the specific responsibility of computer program design, preparation, and development.

At the time that this joint program was initiated (March 1962), DSE was engaged in an overall study of the system requirements for orbital rendezvous between two space vehicles and 496L SPO was engaged in the development of computer programs for tracking and cataloguing existing and future satellites. One phase of the DSE study, known by the code-name COSINE (Co-orbital Satellite Inspection Evaluation), was of common interest to both the RCAF and the USAF and as such was undertaken as a joint study. This study concerns itself primarily with the problems of initial detection, tracking, orbit determination, and prediction of future satellite position.

The experience of both agencies in their respective fields led to the design and production of a computational system to be used in this study. This document contains a complete description of the system in its present form. As a continuing effort, the system and the documentation will be revised and augmented in the coming months. This philosophy accounts for the notebook presentation of the program manual.

The title Simulation Error Analysis System has been assigned to the computer programs written by Aeronutronic for the 496L SPO. Operationally, this system will be used in the COSINE 5 phase of the DSE study. It greatly intensifies an earlier program for this phase known as COSINE 5A, and has subsequently been assigned the code name of COSINE 5B. The two designations, Simulation Error Analysis System and COSINE 5B, are synonymous and differ only because of the historical background in the system development.

The basic philosophy and gross features of this programming system are presented in the remainder of this section. Subsequent sections provide the user with the program operating instructions, major program unit descriptions, executive program description, subroutine specifications, data block and card formats, and system test cases.

The Simulation Error Analysis System has been developed for the Philco 2000 computer, making extensive use of the Philco 2000 Operating System (SYS).

1.1 PROGRAMMING SYSTEM

This document describes the programming system fulfilling the requirements of the Simulation Error Analysis System. The primary requirement of the Simulation Error Analysis System is to produce a statistical analysis of the errors in predicting satellite position and velocity which arise because of observational inaccuracies. The fulfillment of this specification requires the performance of a comprehensive set of functions, many of which are equally as essential as the primary result. A programming system providing versatility in performing these functions is therefore a fundamental tool for the execution of such analysis. Consistent with this consideration, the programming system concept adopted utilizes executive programs linking individual program units in automatic sequences. A total of eight sequences utilizing combinations of seven program units comprise the total system to date. This modular philosophy thus provides the desired system flexibility which will allow the addition of new sequences and modification of existing sequences in the future. The first program sequence fulfills the above primary requirements while seven additional program sequences provide the necessary versatility for performance of the required secondary functions.

The study objectives that may be achieved with this modular design system include:

(1) An investigation of the accuracy with which the orbital elements of a satellite can be established for any given network of ground based sensors and given performance characteristics of each sensor in the network.

(2) An assessment of the effectiveness of sensor networks and the extent to which any one sensor in the network can affect orbital elements of a satellite established by the network as a whole.

(3) A study of the effect of smoothing time between successive observations of a sensor on the accuracy of orbit determination.

(4) An evaluation of the relative merits of different types of sensors in which measurements are used to establish the orbit of a satellite.

(5) The determination of the requirements for performance characteristics (measurement accuracies, coverage limits, etc.) of a sensor or a network of sensors for different levels of accuracy in orbit determination. It is important to determine the measurement accuracy beyond which further improvement does not significantly affect the accuracy of orbit determination.

(6) The establishment of corresponding levels of measurement accuracy among different types of sensors to ensure given levels of accuracy of orbit determination.

(7) The determination of the extent to which the number of observations taken by a given sensor affects the quality of the computed orbit and whether there is a point beyond which additional observations do not contribute a significant improvement in the accuracy of a computed orbit.

(8) An investigation of the effect of observing selected segments of a pass over a sensor for use in providing observation schedules for sensors which are required to observe many satellites in orbit at the same time.

(9) An investigation of the existence of any correlations in the errors present in the computed orbital elements.

(10) The frequency distributions of the errors in orbital elements that can be tested for goodness of fit with assumed distribution laws. If consistency with any given law can be established, analytical methods may then be used to obtain the expected errors in orbital elements from a knowledge of the sensor performance characteristics.

(11) An evaluation of the accuracy with which a satellite position can be predicted at some time after the last sensor observation was possible. This is an important requirement in some military and scientific operations (such as orbital rendezvous).

1.2 PROGRAM SEQUENCES

The program sequences consist of various combinations of the following seven program units:

Acquisition Time Intervals	AQTINT
Observation Simulation	OBSIM
Monte Carlo	MOCARO
Differential Correction	SGPDC
Element Error Distribution	DCDIST
Error Prediction	ERP
Prediction Distribution	ERPDIST

A detailed description of each program unit is presented in Section 3. A diagram schematically representing all of the program sequences with corresponding program units is presented in Figure 1-1, which also contains a brief summary of each program unit function.

The information processing steps for all sequences are illustrated by Figure 1-1. Figure 1-2 is an expansion showing the outputs of individual program units. Both figures specify the entrances and exits for each of the sequences 1 to 8, indicating the program units utilized by each sequence. These figures illustrate the application for each sequence.

The information processing steps in Sequence 1 are as follows:

(1) After Simulation Error Analysis (SEA 1) is loaded in core, the program begins by processing all input case data and loading it in the required core locations. The simulated sensor's observational accuracies represented by fixed values of standard deviations and bias errors are stored in core location SGBLOC. The parameters used to simulate the sensor observing patterns are stored in location IBLOC. Sensor coordinates are loaded in core location SBLOC. The true satellite orbital elements are loaded into location EBLOC.

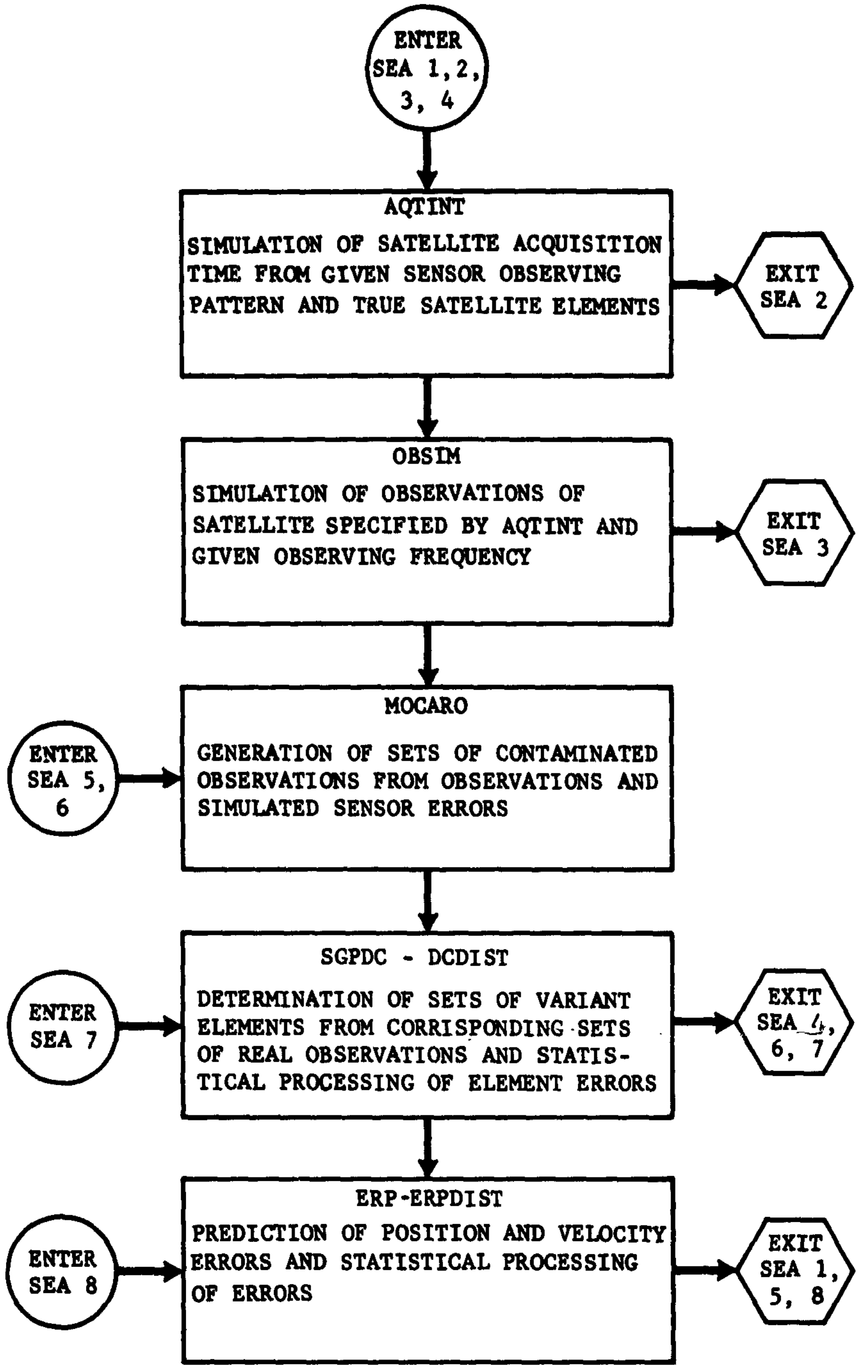

ENTER SEA 1, 2, 3, 4
AQTINT
SIMULATION OF SATELLITE ACQUISITION TIME FROM GIVEN SENSOR OBSERVING PATTERN AND TRUE SATELLITE ELEMENTS
EXIT SEA 2
OBSIM
SIMULATION OF OBSERVATIONS OF SATELLITE SPECIFIED BY AQTINT AND GIVEN OBSERVING FREQUENCY
EXIT SEA 3
ENTER SEA 5, 6
MOCARO
GENERATION OF SETS OF CONTAMINATED OBSERVATIONS FROM OBSERVATIONS AND SIMULATED SENSOR ERRORS
ENTER SEA 7
SGPDC - DCDIST
DETERMINATION OF SETS OF VARIANT ELEMENTS FROM CORRISPONDING SETS OF REAL OBSERVATIONS AND STATISTICAL PROCESSING OF ELEMENT ERRORS
EXIT SEA 4, 6, 7
ENTER SEA 8
ERP-ERPDIST
PREDICTION OF POSITION AND VELOCITY ERRORS AND STATISTICAL PROCESSING OF ERRORS
EXIT SEA 1, 5, 8

FIGURE 1-1 SIMULATION ERROR ANALYSIS SYSTEM

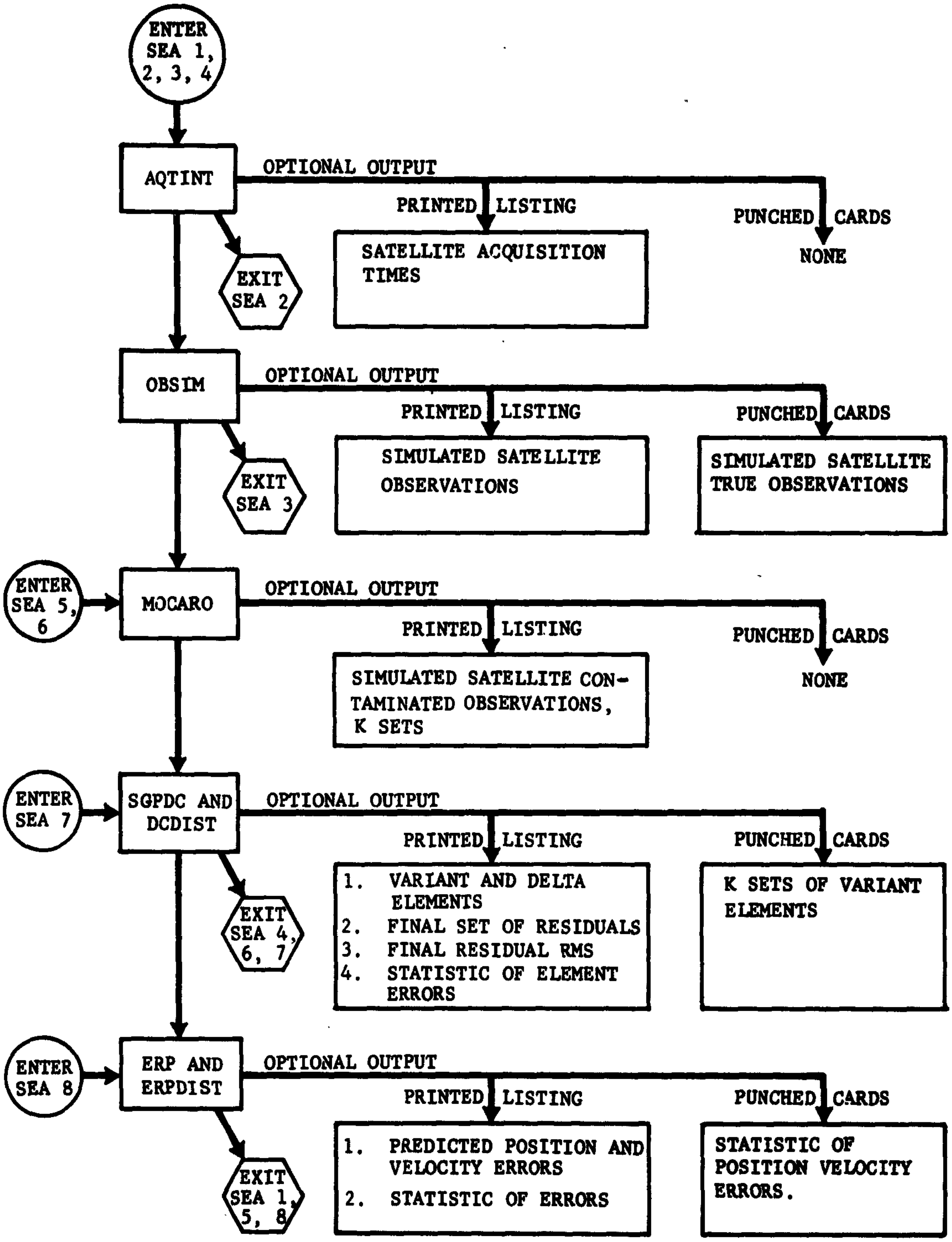


FIGURE 1-2 SIMULATION ERROR ANALYSIS OUTPUT OPTIONS

(2) The data (EBLOC, SBLOC, IBLOC) are processed by the AQTINT Program Unit to provide the times when the satellite is observed by the sensors. These acquisition times are stored in location TBLOC.

(3) Using the data (TBLOC, EBLOC, SBLOC) the OBSIM Program Unit provides a set of simulated observations which are loaded in the Observations block, OBLOC.

(4) The MOCARO Program Unit provides K sets of contaminated observations utilizing the sensor accuracy data previously stored in SGBLOC. A Monte Carlo technique is used to generate random sensor errors from the normal distribution of errors with the given fixed standard deviation. These random errors and given bias errors are added to the simulated observations, thus contaminating them in order to represent actual sensor measurements. The contaminated observations are stored in the contaminated observations block, COBLOC.

(5) Residuals in observations (observed minus computed) are formed and a differential correction of the true orbit elements is performed. Computed observations are those obtained by representing the true satellite orbital elements in EBLOC. The Differential Correction Unit, SGPDC, determines a new set of elements which best fit the observations according to a least squares of residuals method. A rejection option is included to reject large observation residuals on an absolute basis and also on a relative basis. The contamination and correction procedure is repeated for the K sets of observations and produces K sets of variant elements which are then stored in the variant elements block, VEBLOC.

(6) Errors in elements, variant minus true, are computed and a statistical analysis is performed by the DCDIST Program Unit. For each element error, the minimum, maximum, mean, and standard deviation is computed and a frequency distribution performed.

(7) The position and velocity vectors of the satellite are predicted by the ERP Program Unit at selected points from each of the variant element sets. Errors are computed relative to the input element set for each of the position and velocity coordinates and the vector magnitudes and are stored in the delta position and velocity block, DPVBLOC.

(8) Two types of statistical analyses are performed by the ERPDIST Program Unit. In one type, the minimum, maximum, mean error, plus the standard deviation are computed and a frequency distribution is performed for each error quantity in DPVBLOC. In the other type, the mean, maximum, and standard deviation of the position and velocity coordinates and the vector magnitudes for selected points in each quadrant of the orbit are treated as error quantities. Their respective minima, maxima, means, and standard deviations are computed for each of the four quadrants of the orbit. This latter feature provides the trend in each quantity over a segment of the orbit; for instance, it supplies the mean of the mean error in position over one quarter of a revolution.

Sequence 2 (SEA 2) enables one to obtain only the acquisition times as the sequence begins with AQTINT and terminates at its completion. Sequence 3 (SEA 3) will continue through OBSIM, terminating after computing the observational quantities corresponding to the times generated.

Sequence 4 (SEA 4) performs a function similar to SEA 1, with the exclusion of the error prediction. SEA 4 fulfills the objective of an analysis where only the element error statistics are required and no information concerning the position and velocity prediction is desired.

Sequence 5 (SEA 5) accepts elements, observations, and standard deviations and performs the specified number of Monte Carlo, Differential Correction loops. It then continues into DCDIST which produces a statistic of errors in the classical elements at epoch. Finally, ERP predicts position and velocity errors at specified points which are statistically represented by ERPDIST.

Sequence 6 (SEA 6) is similar to SEA 5 except that it excludes the ERPRED functions. ERPRED incorporates the functions of both ERP and ERPDIST.

SEA 3, combined with SEA 5 or SEA 6, provides for an analysis similar to SEA 1 and SEA 4, respectively; however, with the additional capability of selecting particular observations for the error analysis. SEA 3 performs the satellite simulation, producing the observations in punched card form. The selection of these observational data, fulfilling required experimental conditions, provides input to SEA 5 or SEA 6.

Sequence 7 (SEA 7) is SGPDC run with one set of "real" or contaminated observation input. The orbital elements are corrected at epoch and output. At the completion of the correction, the sequence is terminated. SEA 7 is mainly reserved for correcting actual satellite orbits.

Sequence 8 is an ERPRED run only, with the one set of true elements plus N sets of variant elements input ($N \leq 100$). SEA 8 provides the capability of repeatedly performing a variety of prediction analyses on a given group of variant element sets produced by any of sequences SEA 1, 4, 5 and 6.

1.3 SIMULATION CAPABILITIES AND LIMITATIONS

The computational model of satellite motion is based on the theory of two-body motion, with the optional provision of including the effects of the secular perturbations on the orbital elements produced by the equatorial bulge of the earth and atmospheric drag.

The control of the simulation of the sensor observing pattern is provided through a specification of desired limits of the initial acquisition elevation angle and the final acquisition elevation angle. Observations from any of the sensors in the simulated network are specified as any combination of angles (right ascension and declination, or azimuth and elevation angle), range, and range rate. The simulation of observations may be constrained to generate only those observations within a range envelope or those observations consistent with a satellite illumination criterion, representative of the sensor capability. The system has the capacity of simulating a sensor network containing up to 100 different stations and the capability of processing up to 2000 observations (500 time points) for a single satellite simulation.

The major restrictions of the programming system require that:

(1) Satellite orbital eccentricity, $e < 1$.
(2) Satellite orbital inclination, $i \neq 0$.
(3) Satellite orbital period must not be the period of earth rotation.

SECTION 2

SYSTEM OPERATION

The modular design of the Simulation Error Analysis System allows the user the freedom of calling for any of the acceptable sequences. The program units and the associated executive routines, subroutines, and storage assignments are compiled in relocatable binary format and are loaded into core at the execution of one of the sequences. For ease of handling and for a saving in machine time, a particular sequence may be compiled onto the user's designated sequence tape. This disencumbers future operation of the particular sequence, eliminating the time consuming tape searching on the SEA tape (Simulation Error Analysis tape) which is necessary in compiling a sequence.

Thus, there are two modes of operation: one of compiling the desired sequence which may be combined with running the sequence, and one of sequence operation using a previously compiled sequence. The latter operation is executed by submitting a standard SYS job setup resulting in the complete sequence being brought into core by the TAC II loader.

2.1 SEQUENCE COMPILATION INSTRUCTIONS

This paragraph describes the use of SYS service routines to generate a sequence as one program to be retained for future SEA runs. All SEA programs are compiled in relocatable binary format (REL) using symbolic references for intraprogram communication. When more than one REL program is loaded into core memory, the TAC II loader* provides the necessary addresses in those instructions referencing symbolic locations

*See Philco writeup on TAC II and TACII loader.

outside of the original compilation. Therefore, after loading the multiple units comprising a SEA sequence, core will appear as it would if the units had been compiled as one program.

The actual loading of a sequence is accomplished by running a standard SYS job setup with one REL load card for each program comprising the sequence. The last program to be loaded must be EXEC. This REL card has the additional parameters LIST and SUBS. The LIST parameter signals the loader to prepare an edited listing on the output tape (5), giving the octal core address assigned to each symbolic location which was cross referenced between program units. This cross reference aids the programmer in eliminating problems that arise while attempting to run the sequence. The SUBS parameter signals the loader to bring the necessary Philco subroutines into core from the SYS tape (7). These subroutines are also in REL TAC format. Their locations are made available to the sequence programs by symbolic locations in the same manner in which the program units communicate with each other.

After all units have been loaded, core memory will contain the complete sequence in a form suitable for running. The SYS segmentation aids* are then used to write the complete sequence on the Sequence Tape in RPL (Running Program Language) format. In general, this will be a separate tape, but the sequence may be written back onto the SEA tape in order to conserve tape usage. The segmenting routine positions the specified tape to the end of current information, writes the new program, and rewrites the "end of data" indicator (a block of Z's).

The reason for loading the sequence once, and writing it on tape in RPL format, is to save computer time when running. It takes approximately 3 minutes to load Sequence 1 composed of 10 REL units. This time is independent of tape searching time which will be variable depending on the length of the SEA Tape. After the sequence in on the SEA tape in RPL format, it takes less than 10 seconds to load. Therefore, it is to the user's advantage to save the sequence via segmentation aids.

The description below outlines the specific procedure for compiling program sequences.

(1) Tape Unit Designations

Set the tape units with the following fixed assignments:

Logical

1. SYS Tape - no write Ring
2. Scratch tape, write Ring

*See Philco R and D Note Number 28

3. Must have a tape mounted, with or without a write Ring
4. Must have a tape mounted, with or without a write Ring
5. Output tape, write Ring
6. Must have a tape mounted, with or without a write Ring
7. Philco Subroutine tape, no write Ring
8. Input tape, no write Ring

Variable Tape

The SEA tape containing the REL programs will be on one unit, and an optional output tape, other than the SEA tape, can be mounted, (notes c and d below explain these assignments).

(2) Job Deck Setup

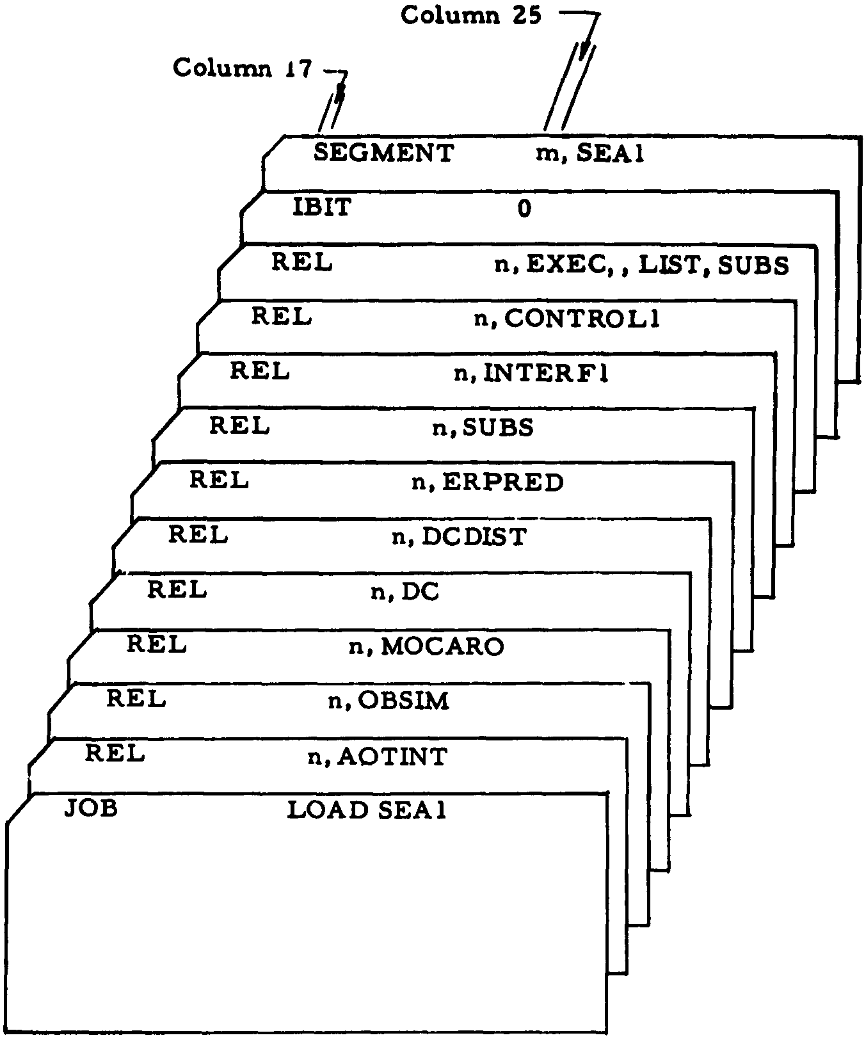
One or more of the job decks presented in Figure 2-1 through 2-8 are used as input to SYS by transferring the card data to magnetic tape and using it as the input tape (logical 8).

(3) Restrictions and Variations

The job decks shown are samples and must adhere to the following restrictions:

(a) Program ID's - With new compilations of the program units, the ID's will be changed, i.e., AQTINT may be changed to AQTINTA. Therefore, it is necessary to see that the current ID is given on the REL card.

(b) Sequence ID's - Because of changes in the program units, it may be necessary to recompile a sequence. If so, the sequence ID must be changed. For example, if Sequence 1 was compiled with the ID on the segment card given as SEQI and a program unit is changed, e.g., AQTINT to AQTINTA, it is necessary to recompile SEQI calling for AQTINTA and also to rename the new compilation, e.g., SEQIA. At run time the



```
SEGMENT       m, SEA1
IBIT          0
REL           n, EXEC,, LIST, SUBS
REL           n, CONTROL1
REL           n, INTERF1
REL           n, SUBS
REL           n, ERPRED
REL           n, DCDIST
REL           n, DC
REL           n, MOCARO
REL           n, OBSIM
REL           n, AOTINT
JOB           LOAD SEA1
```

FIGURE 2-1. SEQUENCE 1 COMPILATION
JOB DECK

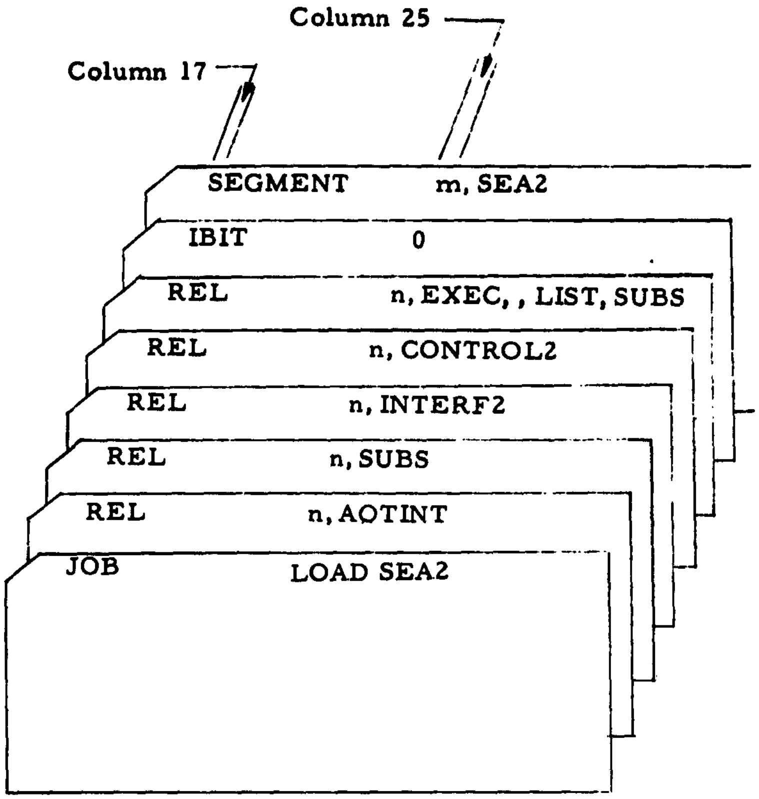
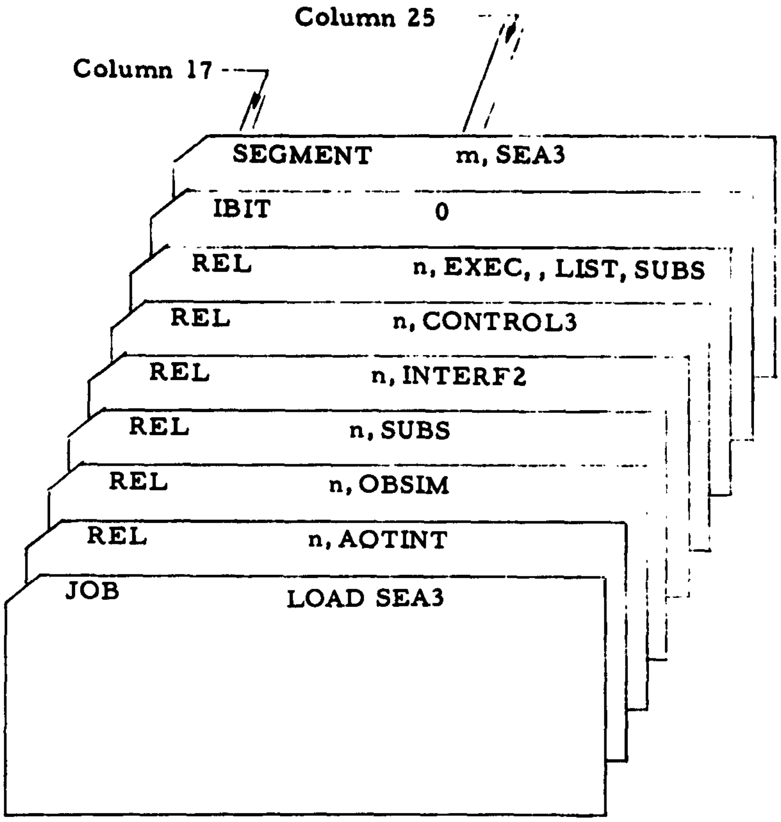
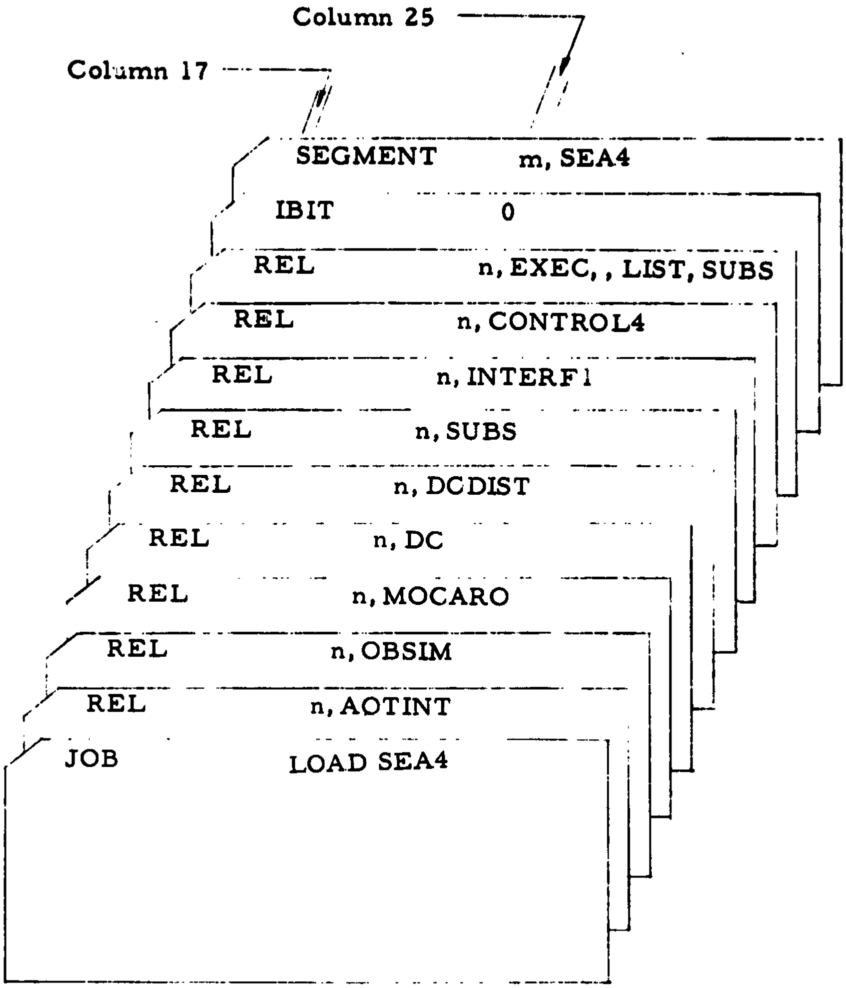


FIGURE 2-2. SEQUENCE 2 COMPILATION JOB DECK



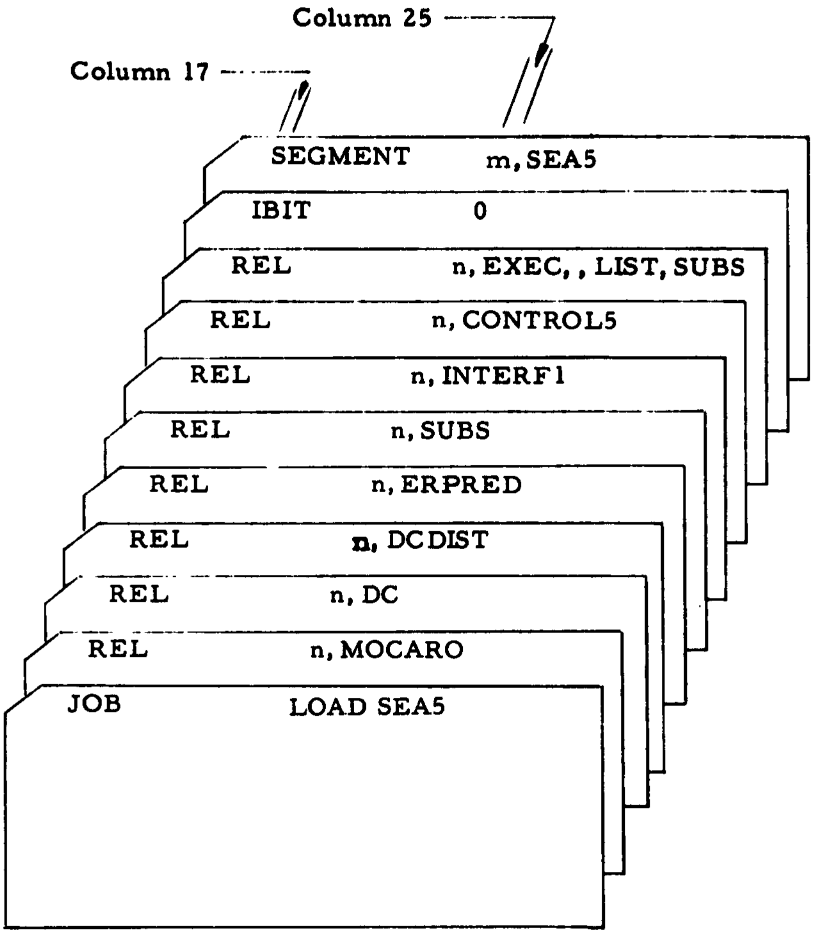
```
SEGMENT   m, SEA3
IBIT      0
REL       n, EXEC, , LIST, SUBS
REL       n, CONTROL3
REL       n, INTERF2
REL       n, SUBS
REL       n, OBSIM
REL       n, AOTINT
JOB       LOAD SEA3
```

FIGURE 2-3. SEQUENCE 3 COMPILATION JOB DECK



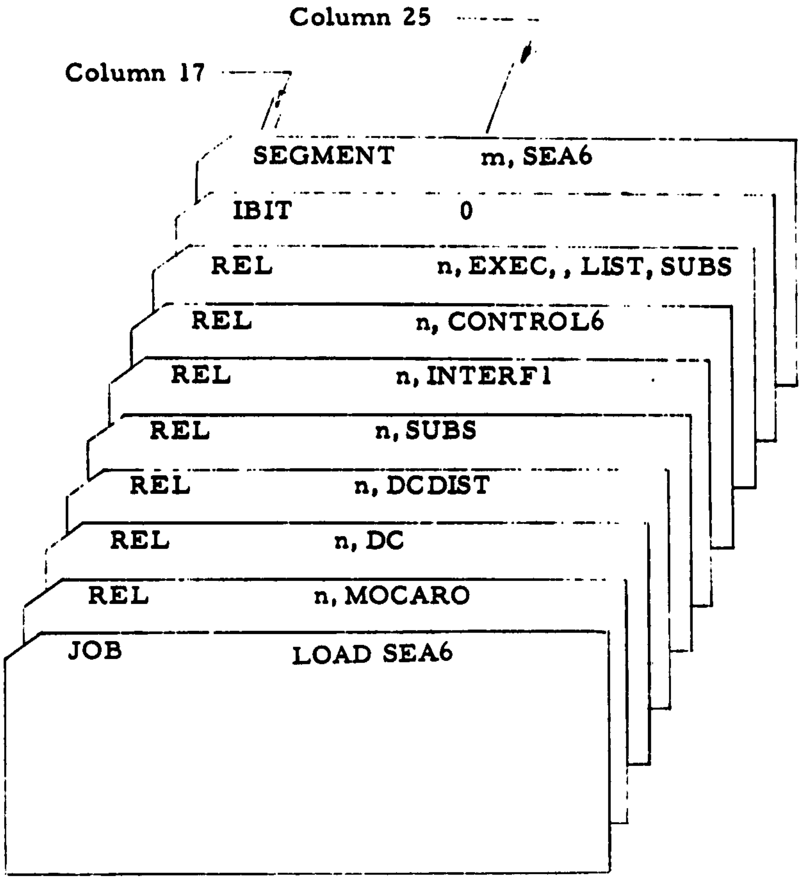
```
SEGMENT         m, SEA4
IBIT            0
REL             n, EXEC, , LIST, SUBS
REL             n, CONTROL4
REL             n, INTERF1
REL             n, SUBS
REL             n, DCDIST
REL             n, DC
REL             n, MOCARO
REL             n, OBSIM
REL             n, AOTINT
JOB             LOAD SEA4
```

FIGURE 2-4. SEQUENCE 4 COMPILATION JOB DECK



```
SEGMENT  m,SEA5
IBIT     0
REL      n,EXEC,,LIST,SUBS
REL      n,CONTROL5
REL      n,INTERF1
REL      n,SUBS
REL      n,ERPRED
REL      n,DCDIST
REL      n,DC
REL      n,MOCARO
JOB      LOAD SEA5
```

FIGURE 2-5. SEQUENCE 5 COMPILATION JOB DECK



```
SEGMENT        m, SEA6
IBIT           0
REL            n, EXEC,, LIST, SUBS
REL            n, CONTROL6
REL            n, INTERF1
REL            n, SUBS
REL            n, DCDIST
REL            n, DC
REL            n, MOCARO
JOB            LOAD SEA6
```

FIGURE 2-6. SEQUENCE 6 COMPILATION JOB DECK

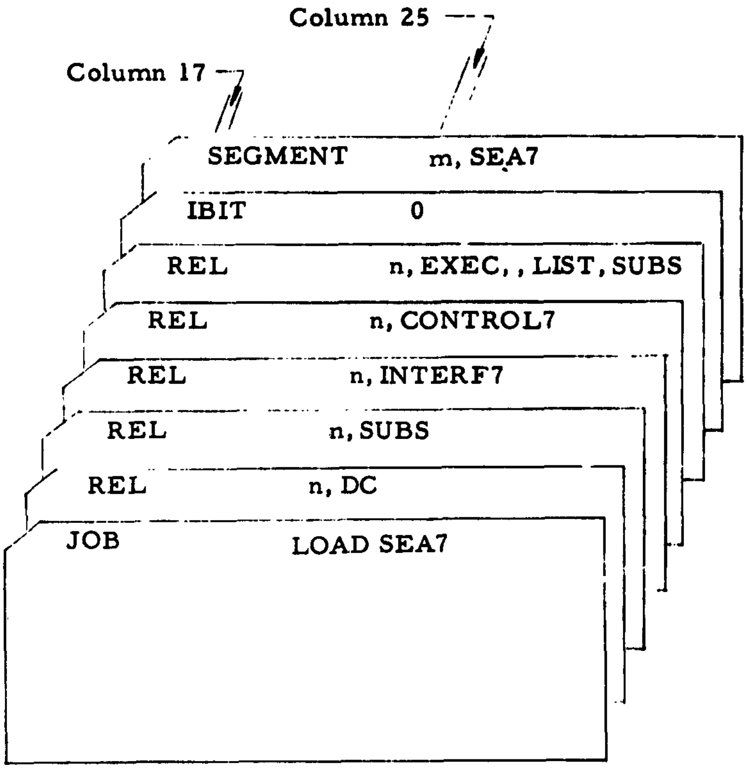
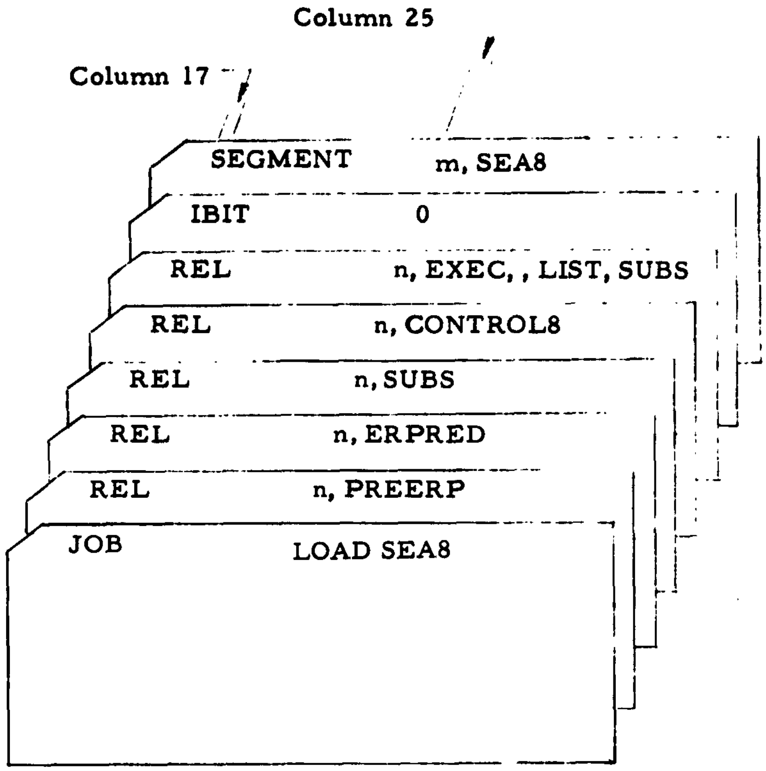


FIGURE 2-7. SEQUENCE 7 COMPILATION
JOB DECK



```
SEGMENT  m, SEA8
IBIT     0
REL      n, EXEC, , LIST, SUBS
REL      n, CONTROL8
REL      n, SUBS
REL      n, ERPRED
REL      n, PREERP
JOB      LOAD SEA8
```

FIGURE 2-8. SEQUENCE 8 COMPILATION JOB DECK

RPL card must also be changed to agree with the new sequence ID. (This ID has a maximum of 8 characters). Care must be taken that no program, either an REL unit or a segmented sequence, be on the same tape twice with the same ID. This will require machine time to remedy and is very costly, both in dollars and time lost.

(c) SEA Tape Mounting - The SEA tape may be mounted on any unit but 1,2,5,7,or 8, and may or may not have a write ring. The tape number "n" on the REL cards in input must correspond to the logical tape number on which the SEA tape is mounted.

(d) Output Tape Mounting - The output tape for the RPL version of the compiled program (Sequence tape) may be on any tape but 1, 2, 4, 5, 7, or 8. The tape number "m" on the SEGMENT card must correspond to the logical tape number on which the output tape is mounted.

It is admissible to write the compiled sequence onto the SEA tape. In this case, the SEA tape must have a write ring, and cannot be on tape unit 4.

(e) Sentinal Block - The "IBIT O" instruction causes the segmenting program to search the tape for a sentinal block of Z's before writing the program. Therefore, if an "IBIT O" is used, the tape specified in the SEGMENT instruction <u>must</u> have a block of Z's after all good information. If the "IBIT O" is not used, the segmenting program will assume that the tape specified is properly positioned and begin writing the RPL program without prepositioning the tape. This is an extremely dangerous practice, since an error will destroy the tape data. If the tape on which the sequence(s) will be compiled is a scratch tape it is not the SEA tape and i.e., no programs or sequences have been written on it, it is necessary to precede the first sequence compilation job deck with the following job deck:

```
JOB         WRITE      SENTINEL

REWIND      m
                                          m = logical tape no.
WRTSENT     m, Z Z Z Z Z Z Z Z               of output tape.

REWIND      m
```

If only one sequence is to be compiled on a scratch tape, one has the option of replacing the "IBIT 0" card with a "REWIND m". This will cause the sequence to be written on the beginning of the tape followed by a sentinel block; the tape will be positioned at the beginning of the sentinel block.

2.2 SEQUENCE OPERATION

SEA sequence runs are executed under SYS control. This means that standard SYS tape and input deck setups are used. Tapes required for a sequence run are:

Logical

1. SYS Tape
2. Scratch Tape
3. Scratch Tape
4. Sequence Tape

5. Output Tape
6. Scratch Tape
7. Not required
8. Input Tape

9-15. Not required

Sequence output is on Tape 5. Printer or hard copy output is in data select 0. Punched card output is in data select 2 and should be punched in code mode with the "sense control characters" switch set to ignore (Punched Card Controller Switches).

Sequence input is on Tape 8. One job consists of one or more cases on the same SEA sequence, i.e., one job may contain several cases on SEA1 only. Jobs always begin with a JOB card, JOB in Col 17-24, and contain an RPL card, RPL in Col 17-24, calling the sequence desired. Dump cards, DUMP in Col 17-24, are optional and may precede or follow the RPL card. All cards after these are SEA cards and actual input data.

One case in a SEA deck is always preceded by a case card, SEA in Col 17-24, and followed by an ENDDATA card, ENDDATA in Col. 17-24. The last case in a job should be followed by an SYS card, SYS in Col 17-24. In between the case card and the ENDDATA card is the input data consisting of all program control and data cards necessary to run the case. The Job Deck setup required to run sequences 1 to 8 is included in Figures 2-9 through 2-16. Any one case will be subject to the following restrictions:

(1) One element set consisting of two element cards is the maximum allowed. This can be type A, B, C, or D. Type C and D elements must be preceded by the associated sensor card.

(2) For element input type A or B, 100 sensor cards are allowed. For element types C and D, one sensor card is allowed, the sensor associated with the elements.

(3) One hundred simulation parameter cards are allowed. There must be a sensor and a standard deviation card for each different sensor specified on the simulation parameter cards.

(4) One hundred standard deviation cards are allowed.

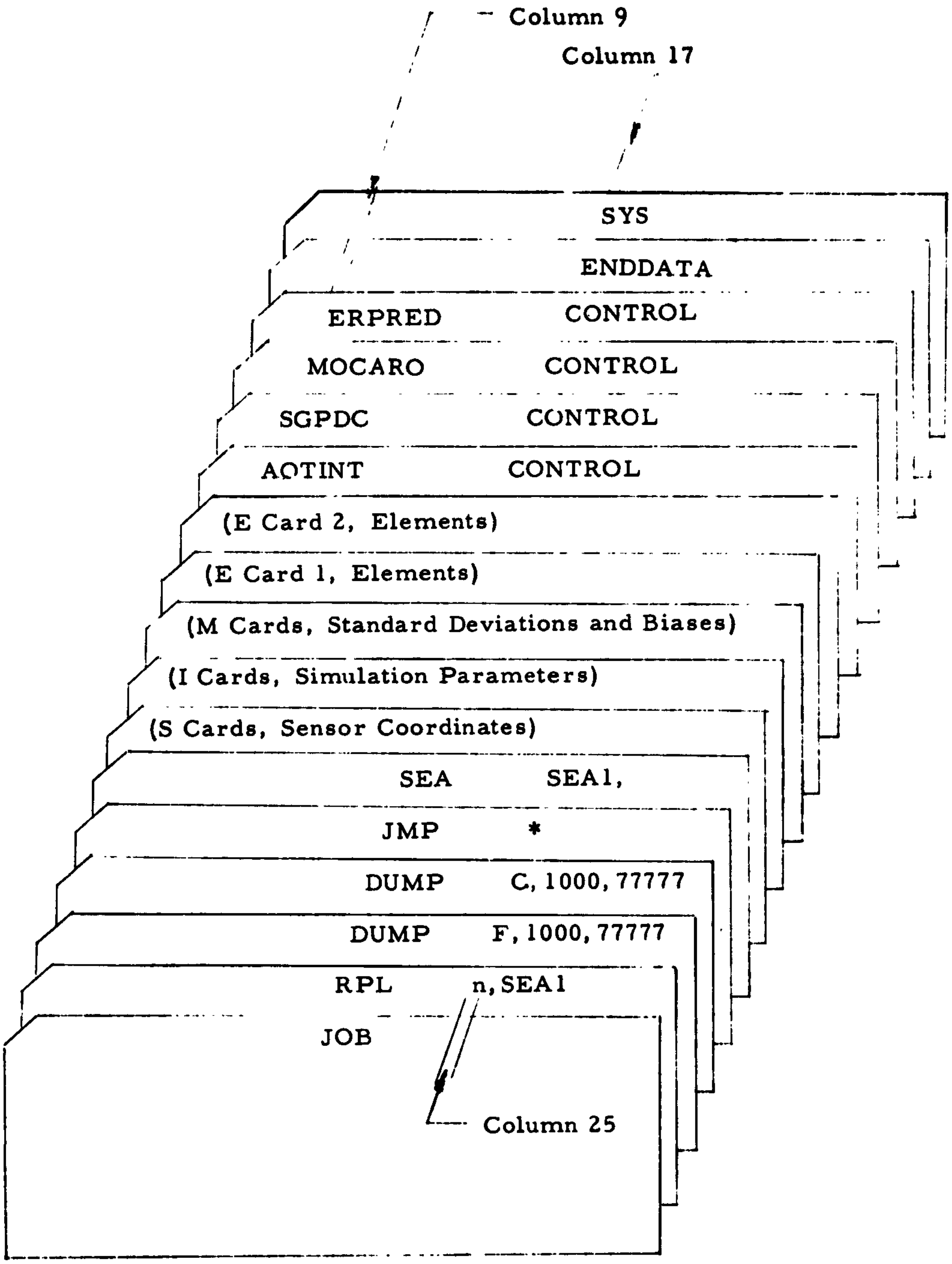


FIGURE 2-9. SEQUENCE 1 RUN JOB DECK

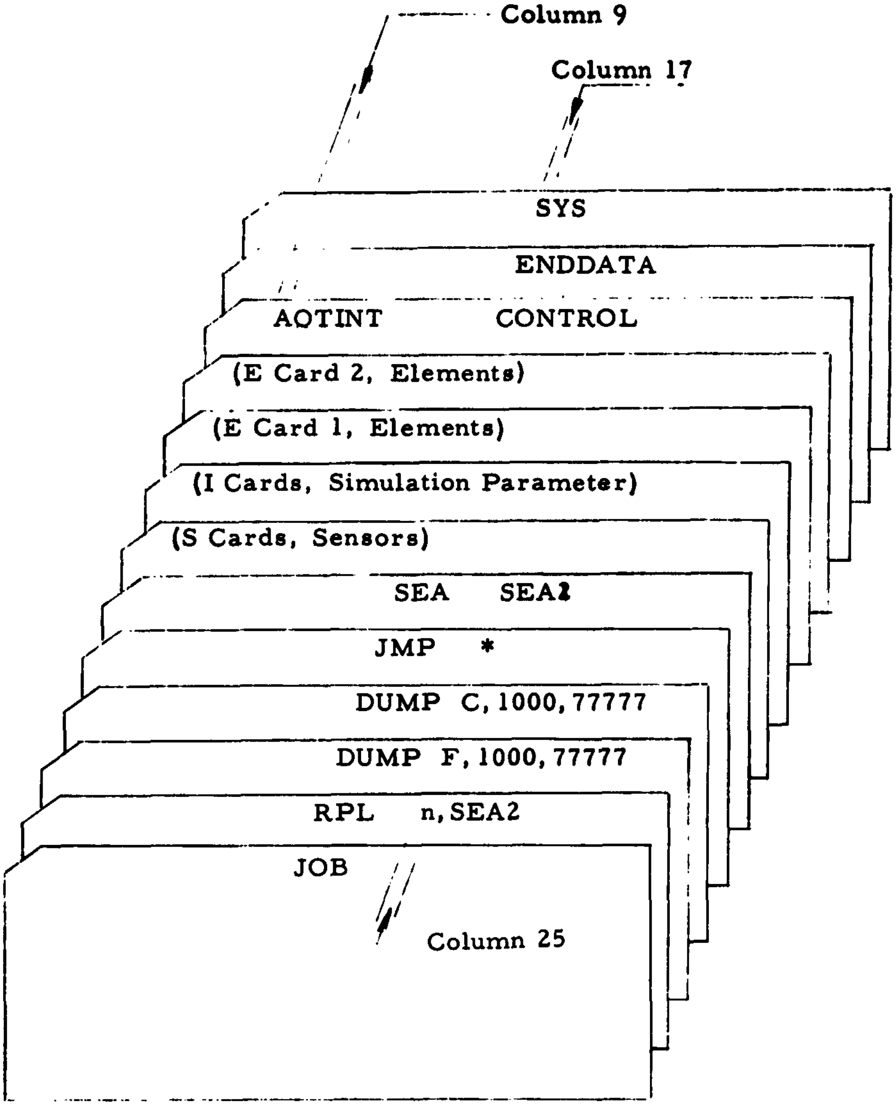


FIGURE 2-10. SEQUENCE 2 RUN JOB DECK

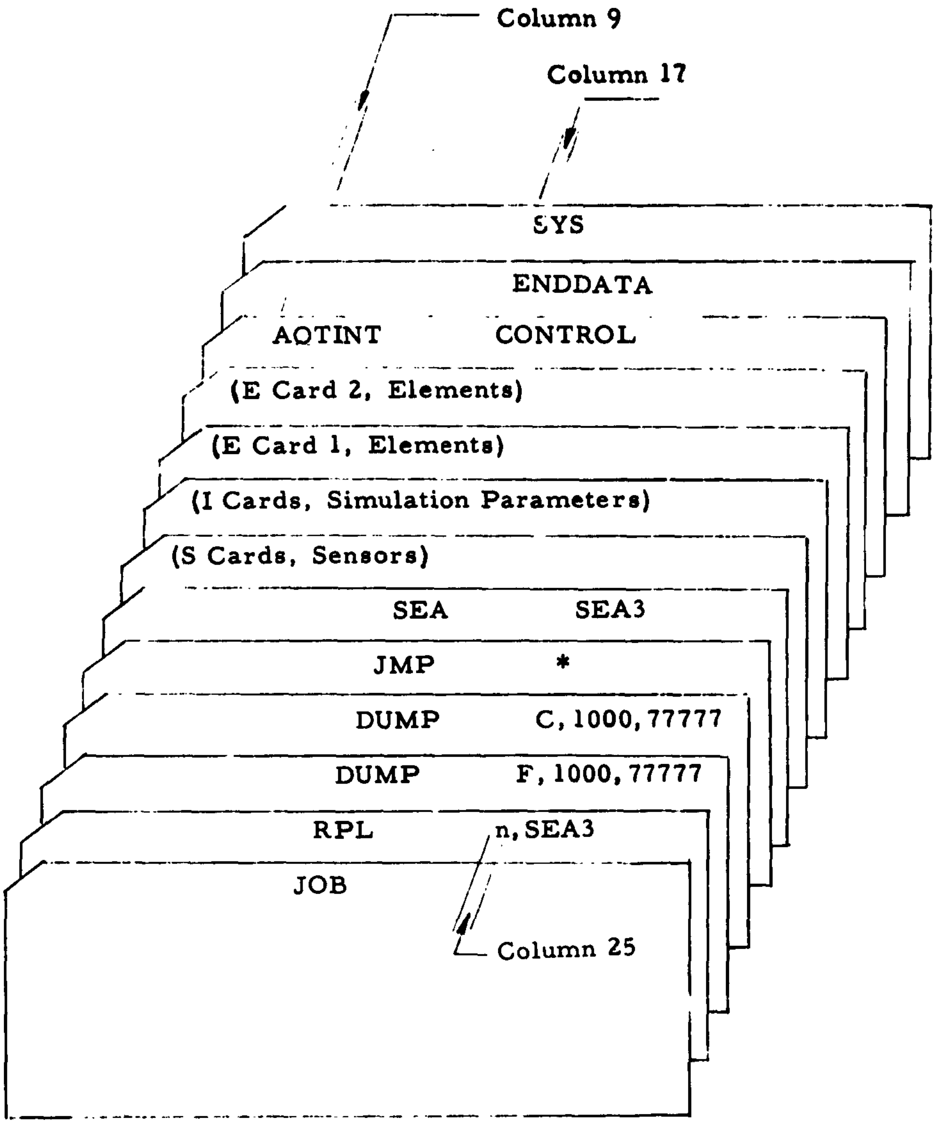


FIGURE 2-11. SEQUENCE 3 RUN JOB DECK

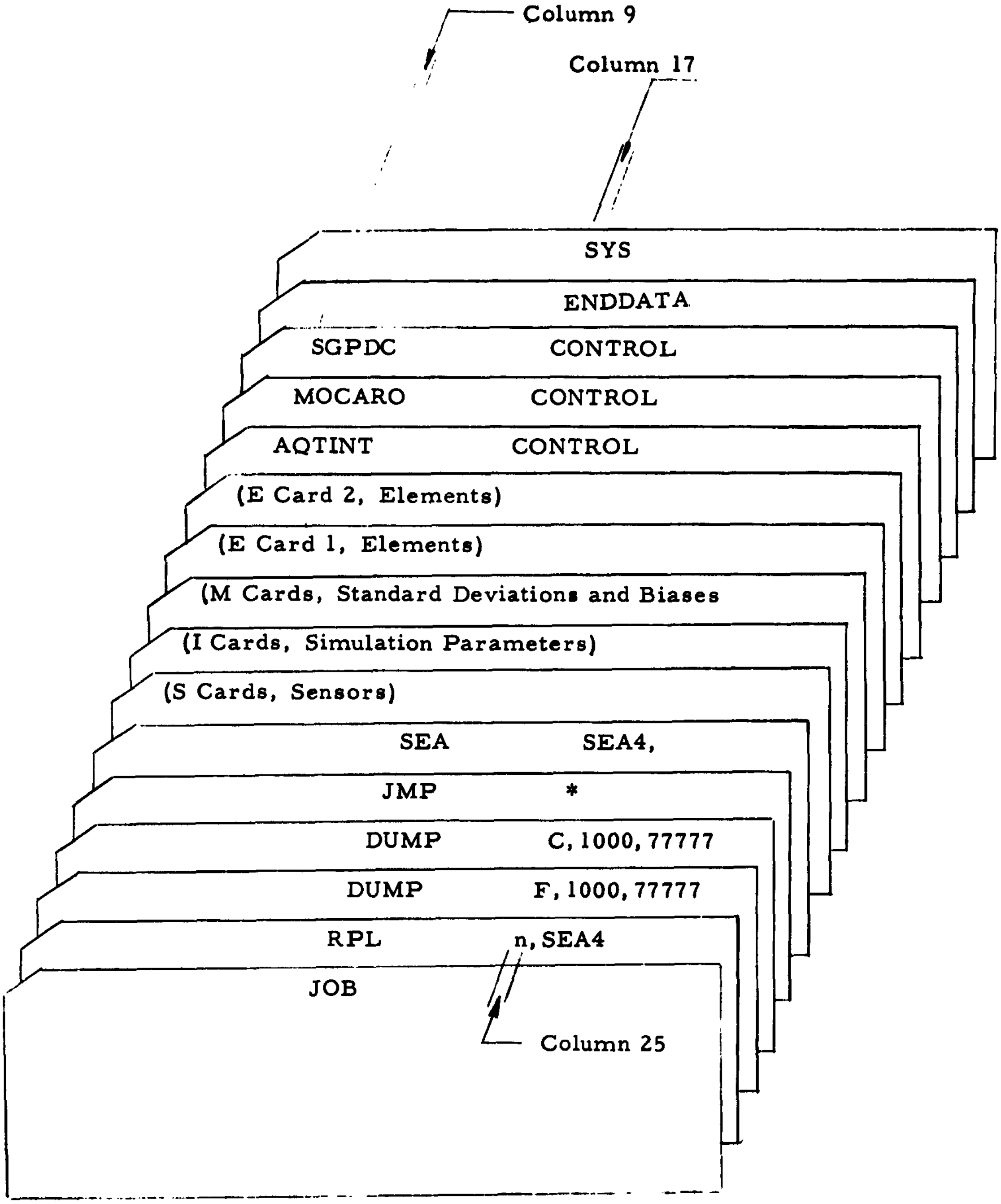


FIGURE 2-12. SEQUENCE 4 RUN JOB DECK

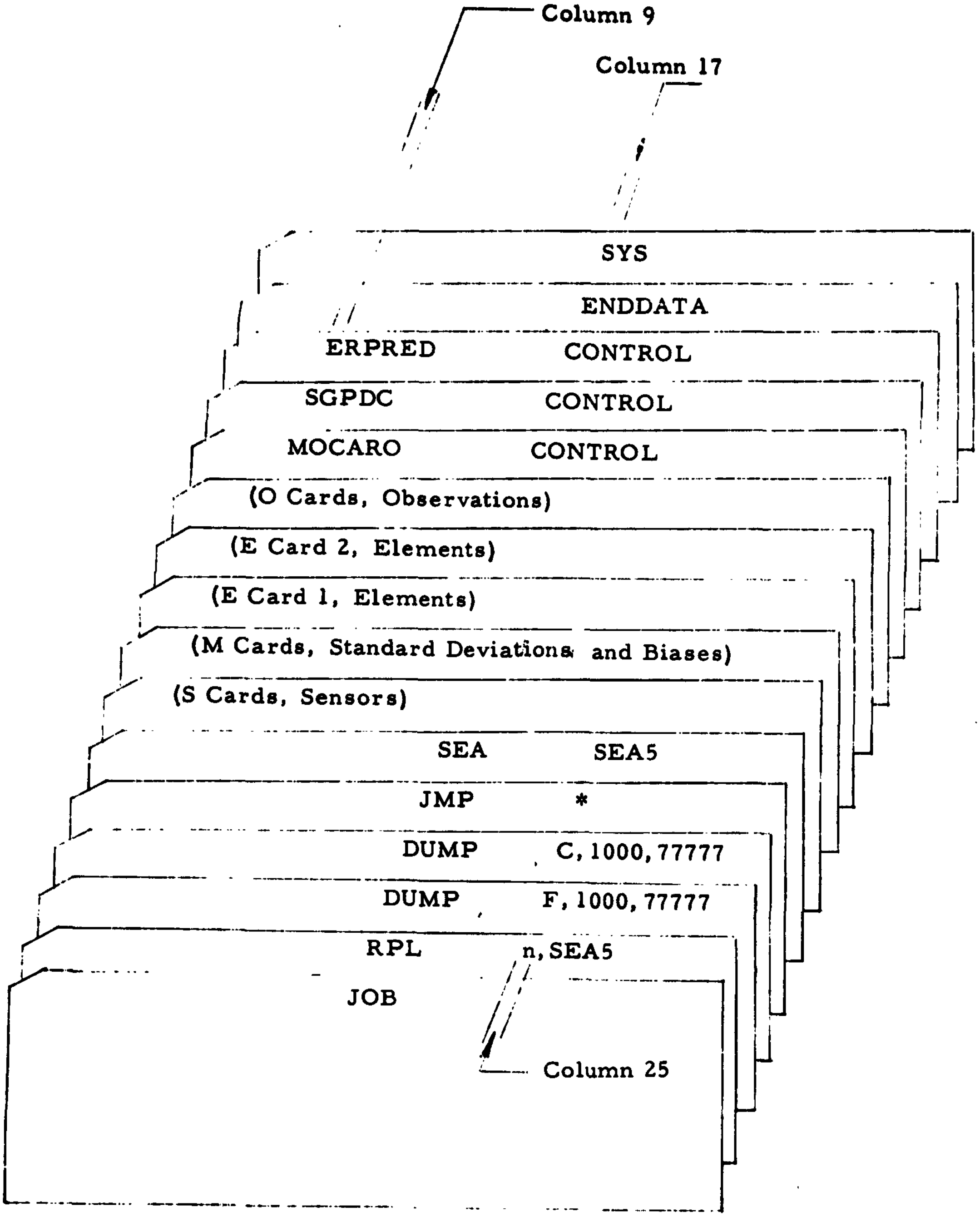


FIGURE 2-13. SEQUENCE 5 RUN JOB DECK

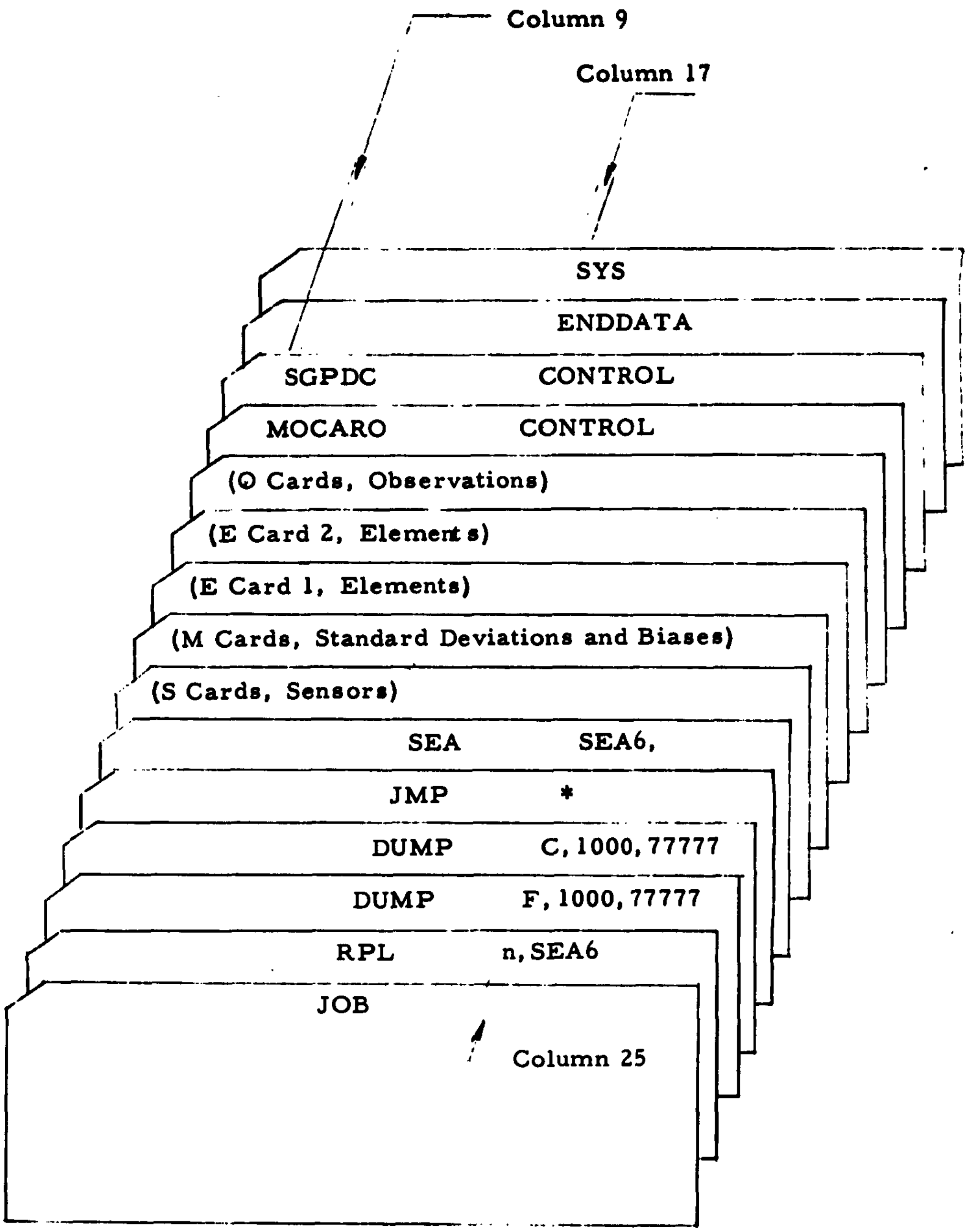


FIGURE 2-14. SEQUENCE 6 RUN
JOB DECK

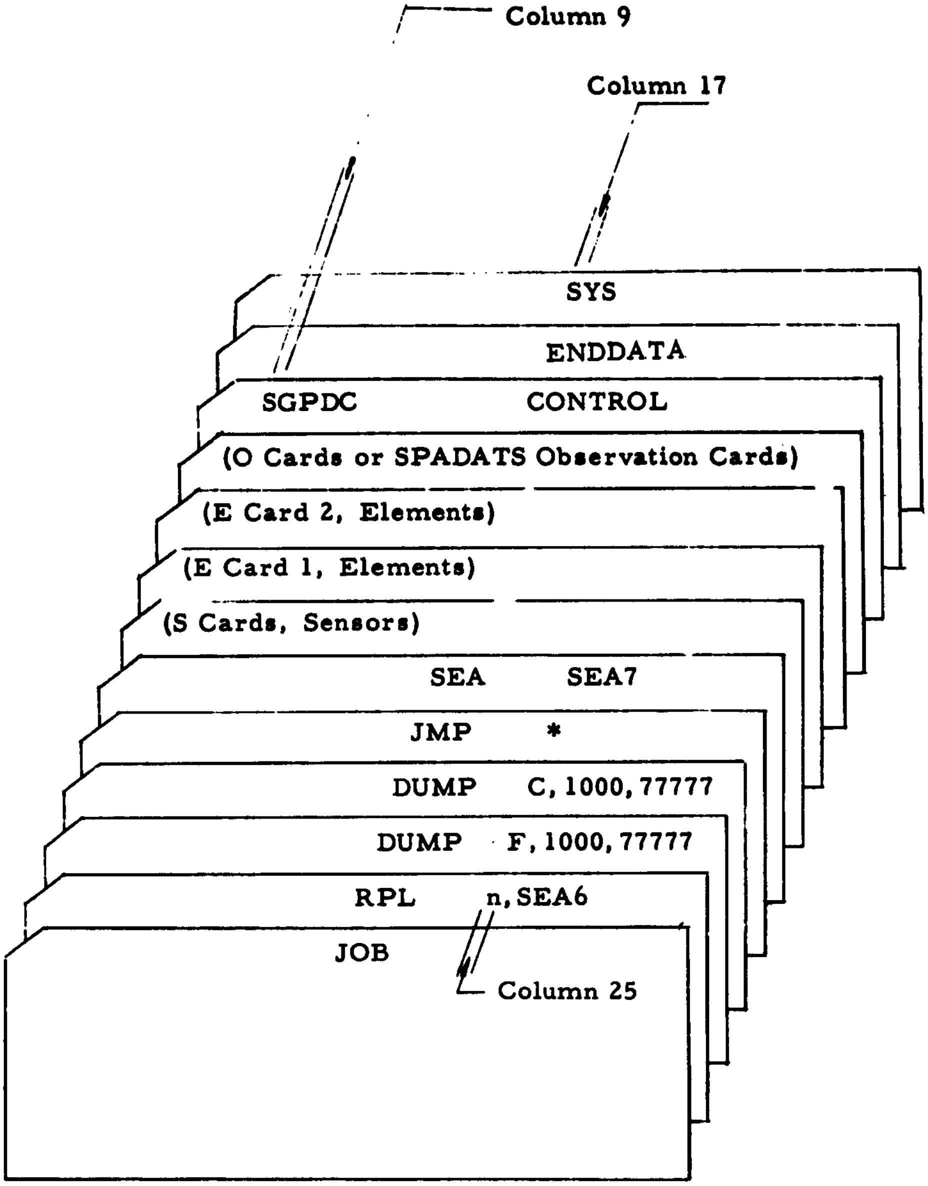
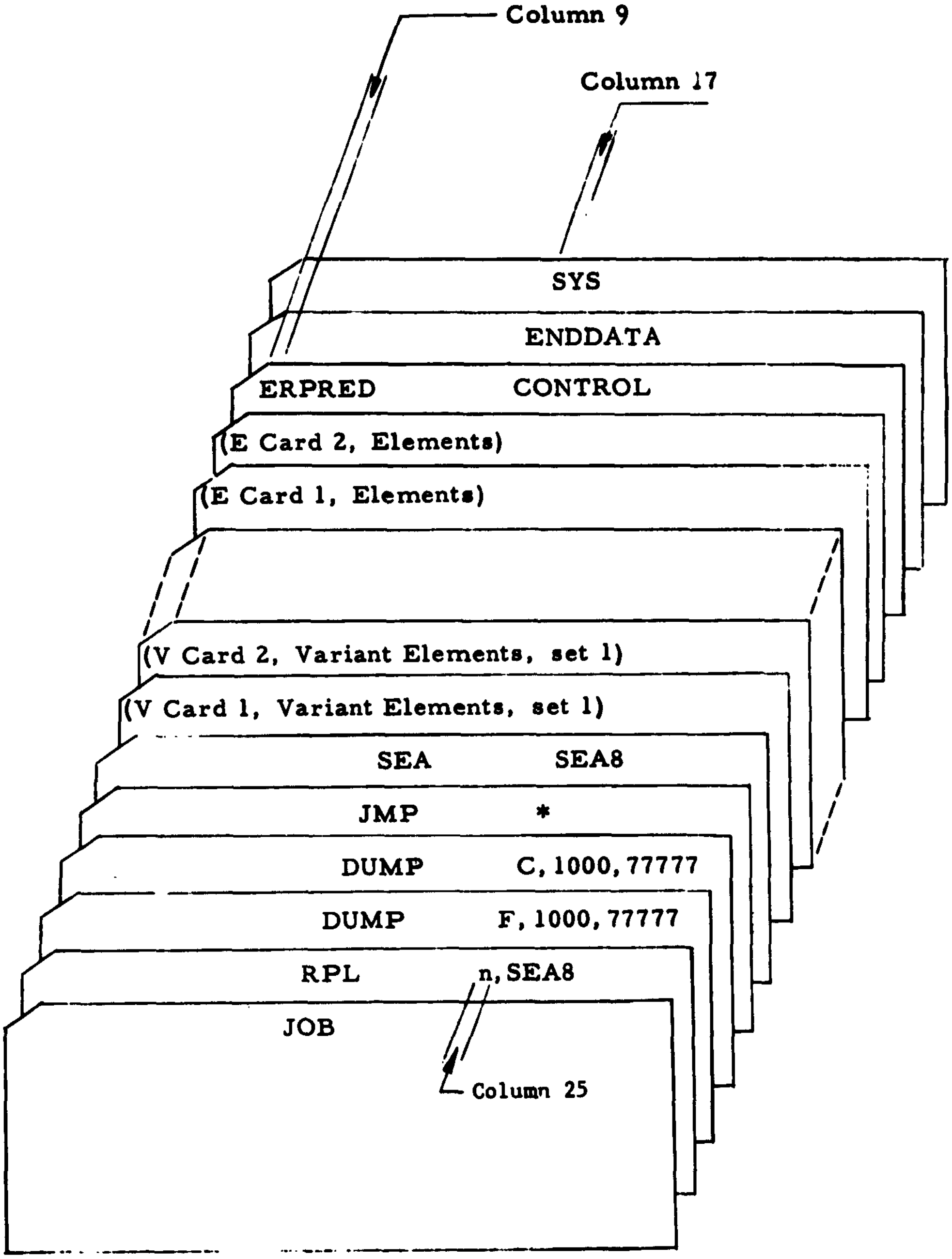


FIGURE 2-15. SEQUENCE 7 RUN JOB DECK



```
                SYS
                ENDDATA
ERPRED          CONTROL
(E Card 2, Elements)
(E Card 1, Elements)
(V Card 2, Variant Elements, set 1)
(V Card 1, Variant Elements, set 1)
                SEA     SEA8
                JMP     *
                DUMP    C,1000,77777
                DUMP    F,1000,77777
                RPL     n,SEA8
                JOB
```

FIGURE 2-16. SEQUENCE 8 RUN JOB DECK

(5) Observations can be standard SPADATS format, 0-9 or Λ in Col 80, or SEA format, 0 in Col 80. If the elements are type A or B, either type may be used. If SPADATS format is used, the elements must precede the observations. If elements are types C or D, only SEA format observations are allowed. In any case, 500 observations is the maximum allowed.

(6) A maximum of 500 variant element sets is allowed. These must be type A or B, and must be preceded by the element, nominal elements, input.

(7) One control card of each specified type should be included.

SECTION 3

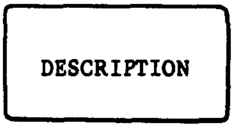
PROGRAM UNITS

The seven major program units comprising the Simulation Error Analysis System are:

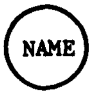
Acquisition Time Intervals	AQTINT
Observation Simulation	OBSIM
Monte Carlo	MOCARO
Differential Correction	SGPDC
Element Distribution	DCDIST
Error Prediction	ERPRED
Prediction Distribution	ERPDIST

This section presents a detailed description of each program unit in terms of general function, input-output data requirements, computational and log:c processes. Also, each program unit is summarized according to system data blocks, associated input cards, and common subroutines used. Reference should be made to Sections 5, 6, and 7 for detailed formats of the respective blocks, cards, and common subroutines.

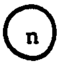
In addition to the program formulation subsections, a schematic representation of each program unit is also exhibited. These flow diagrams serve to augment the description of the computation and logic processes for each program unit. The following definitions of the symbol logic will aid in following these flow diagrams:



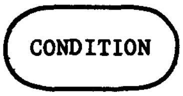
Operation symbol with operation described within the box.



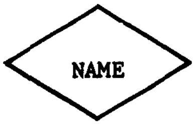
Outside entrance or exit with symbolic location "NAME".



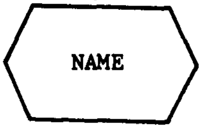
Connector, flow chart reference = n.



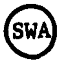
Test or branch - if condition exists, exit to the "yes" path; if condition does not exist, exit to the "no" path. (May be comparison with = vs ≠, < vs ≥, etc).



Common subroutine, "NAME" writeup appearing in Section 5.



Program subroutine, either on flow chart or in program writeup.



Internal program Switch "A" or other designation ("B", "C", etc.) path will vary depending on preset value assigned to the switch.

NAME

Written adjacent to any of the symbols above, this is the symbolic location in the coding associated with that function.

The individual program units are preceded in this section by a description of the initializing process. This is not a program unit per se, but is required prior to entering any of the system sequences. The control of this initializing process lies within the executive function of the sequences. Initialization is executed immediately after SYS transfers control to the sequences.

3.1 INITIALIZATION

The computations required before entering any of the program sequences include the conversion of the four input element types (A,B,C,D) to a common type, computation of Greenwich sidereal time at epoch, retrieval of year constants for the sun, and transformation of sensor coordinates to a rectangular form. The system actually contains these computations in the executive subroutines ELMRED and SENCARD and not in a separate initialization unit. The outputs generated by these routines are germane to the program units and therefore are presented in this initialization section.

3.1.1 INPUT-OUTPUT DATA

The input data required by the initializing subroutines include:

(1) Element cards (four types accepted)

Type A (a, e, i, Ω, ω, M_o)

Type B (x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$)

Type C (H_q, e, i, h_c, v_c, ρ_o)

Type D (a, e, i, Ω, ω, h_o)

(2) Sensor cards

The output data produced by the ELMRED subroutine consist of:

(1) EBLOC - system data block containing input elements in M N form, time of epoch, Greenwich sidereal time at epoch, and satellite identification.

(2) Mean longitude of the sun at the beginning of the epoch year (types A and B only).

(3) Corrective constant used in computing the true longitude of the sun (types A and B only).

The output from the SENCARD subroutine produces SBLOC, a system data block containing sensor coordinates in rectangular form.

3.1.2 FORMULATION

Element types A and B require the computation of the Greenwich sidereal time at epoch, θ_{t_o}, by means of

$$\theta_{t_o} = \theta_o + 0.9856472 \text{ (integer)} + 360.9856472 \text{ (FRAC)}, \qquad (3.1.1)$$

where integer is the day number since the beginning of the year and FRAC is the fractional part of the day. θ_o is the Greenwich sidereal time at the beginning of the year, obtained from a table look-up procedure, STLC subroutine. This subroutine also provides $L_{\odot}$, the mean longitude of the sun at the beginning of the year, and C_3, a corrective constant for the same year.

Element type C is arbitrary with respect to epoch, and therefore the conversion of this type is made with the simplifying assumption that $t_o = 0$ and $\theta_{t_o} = 0$. Types C and D depend on station location and require the coordinates of the station to be available before conversion.

The transformation of the geodetic station coordinates to rectangular coordinates is accomplished by the relationships,

$$x_c = (C + H) \cos \Phi \qquad (3.1.2)$$

$$y_c = (S + H) \sin \Phi$$

where

y_c = sensor position above equatorial plane.

x_c = projection on the equatorial plane of the distance from geocenter to sensor.

The quantities C and S are conversion factors based on the International Ellipsoid with flattening, f = 1/298.3, and equatorial radius $a_e = 1$ (6378.15 km),

$$C = \frac{1}{\sqrt{1 - (2f - f^2) \sin^2 \Phi}}, \qquad (3.1.3)$$

$$S = (1 - f)^2 C. \qquad (3.1.4)$$

and

H = altitude above the reference ellipsoid (approximately sea level).

Φ = geodetic latitude.

The output data block, SBLOC, contains the quantities, east longitude, λ_E, - x_c, - y_c, Φ, and H.

The element conversion subroutines compute the M N elements

$$(a_{xN_o}, a_{yN_o}, h_{x_o}, h_{y_o}, h_{z_o}, L_o)$$

from any of the four input types, A, B, C or D.

The conversion for classical elements (ETYPA) is straightforward. First, the semilatus, p, is found from the semimajor axis, a, and the eccentricity, e, by

$$p = a(1 - e^2). \tag{3.1.5}$$

The components of the angular momentum vector, $\underline{h}$, follow, using the right ascension of the ascending node, Ω, and the inclination, i,

$$h_{x_o} = \sqrt{p} \sin \Omega \sin i, \tag{3.1.6}$$

$$h_{y_o} = -\sqrt{p} \cos \Omega \sin i, \tag{3.1.7}$$

$$h_{z_o} = \sqrt{p} \cos i. \tag{3.1.8}$$

The nodal components of the a vector, which depend on e and the argument of perigee, ω, are computed from

$$a_{xN_o} = e \cos \omega, \tag{3.1.9}$$

$$a_{yN_o} = e \sin \omega. \tag{3.1.10}$$

Finally, the mean longitude at epoch is computed using ω, Ω, and the mean longitude at epoch, M_o, from

$$L_o = M_o \pm \Omega + \omega, \quad \begin{array}{l} + \text{ for } h_z \geq 0 \\ - \text{ for } h_z < 0. \end{array} \tag{3.1.11}$$

The conversion for the position and velocity elements (ETYPB) requires a more complex set of formulae, which are necessary to avoid the singularities associated with low eccentricity. The components of angular momentum are obtained directly from the position vector $\underline{r}$ and the velocity vector $\dot{\underline{r}}$ in the vector product

$$\underline{h} = \frac{\underline{r} \times \dot{\underline{r}}}{\sqrt{\mu}} \tag{3.1.12}$$

where the vector product is computed using the input components $\underline{r}(x, y, z)$ and $\dot{\underline{r}}(\dot{x}, \dot{y}, \dot{z})$. The computation of p follows.

$$p = \underline{h} \cdot \underline{h}. \tag{3.1.13}$$

The orthogonal set of unit vectors is then computed

$$\underline{U} = \underline{r}/r, \tag{3.1.14}$$

$$\underline{W} = \underline{h}/\sqrt{p} \tag{3.1.15}$$

$$\underline{V} = \underline{W} \times \underline{U}, \tag{3.1.16}$$

where $\underline{U}$ is the unit vector along $\underline{r}$, $\underline{W}$ is the unit vector normal to the orbit plane in the direction of the angular momentum vector, and $\underline{V}$ completes the right-handed set. The eccentricity is obtained from

$$e^2 = (e \cos v)^2 + (e \sin v)^2, \tag{3.1.17}$$

where

$$e \sin v = \sqrt{p/\mu}\ \dot{r} \tag{3.1.18}$$

and

$$e \cos v = p/r - 1. \tag{3.1.19}$$

Equations (3.1.12) and (3.1.18) utilize the mass function, μ, which is taken as unity in satellite applications. The radial component of the velocity vector, $\dot{r}$, in Equation (3.1.18) is computed from

$$\dot{r} = \frac{\underline{r} \cdot \dot{\underline{r}}}{r} \qquad (3.1.20)$$

The semimajor axis, a, and subsequently the mean motion, n, are then available through Equation (3.1.5) and

$$n = k_e \sqrt{\mu}\, a^{-3/2} \qquad (3.1.21)$$

where k_e is the geocentric gravitational constant. True longitude at epoch, l_o, is computed from

$$l_o = \tan^{-1}\left[\frac{U_y - V_x}{U_x + V_y}\right] \qquad (3.1.22)$$

The nodal components of $\underline{a}$ follow, using

$$\underline{a} = \underline{U}\, e \cos v - \underline{V}\, e \sin v \qquad (3.1.23)$$

which leads to

$$a_{xN_o} = (-a_x W_y + a_y W_x)(1 - W_z^2)^{-1/2} \qquad (3.1.24)$$

$$a_{yN_o} = (-a_x W_x W_z - a_y W_y W_z)(1 - W_z^2)^{-1/2} + a_z (1 - W_z^2)^{1/2} .$$

The difference between the true anomaly, v, and the eccentric anomaly, E, is computed from

$$v - E = \tan^{-1}\left[\frac{\sin (v - E)}{\cos (v - E)}\right], \qquad (3.1.25)$$

where

$$\sin (v - E) = \frac{r}{p}\left[\frac{(e \cos v)(e \sin v)}{1 + (1 - e)^{1/2}} + e \sin v\right] \qquad (3.1.26)$$

and

$$\cos (v - E) = \frac{r}{p}\left[1 - \frac{(e \sin v)^2}{1 + (1 - e)^{1/2}} + e \cos v\right] . \qquad (3.1.27)$$

Combining

$$e \sin E = \frac{r}{p}\sqrt{1 - e^2}\ e \sin v \qquad (3.1.28)$$

with the results of Equations (3.1.22) and (3.1.25), the mean longitude at epoch is obtained from

$$L_o = l_o - (v - E) - e \sin E. \qquad (3.1.29)$$

The ETYPC conversion routine treats the elements defined by a partial set of elements and a description of the pass over the sensor. This partial set of elements includes the altitude of perigee above the reference ellipsoid, H_q, the elevation angle of the satellite as it crosses the meridian of the sensor, h_c, the true anomaly of the satellite as it crosses the meridian of the sensor, v_c, e, and i. The elevation angle at culmination is specified either north or south of the sensor's zenith. The pass over the station is specified either north-to-south or south-to-north. Even though these north/south descriptions become incongruous when the satellite is at the apex or zenith positions, no ambiguity exists in the mathematics.

The magnitude of the radius vector from the geocenter to the satellite at culmination is obtained by

$$r_c = \frac{q(1 + e)}{1 + e \cos v_c} \qquad (3.1.30)$$

where the perigee distance

$$q = 1 + H_q \qquad (3.1.31)$$

The geocentric latitude of the subsatellite point Φ_s (Figure 3-1) is computed from

$$\Phi_s = \Phi \pm \psi_c, \quad \begin{matrix} + \text{ north of zenith} \\ - \text{ south of zenith} \end{matrix} \qquad (3.1.32)$$

where ψ_c is the complement of the sum of the angles $(h_c + \beta_c)$(see Figure 3-2) and

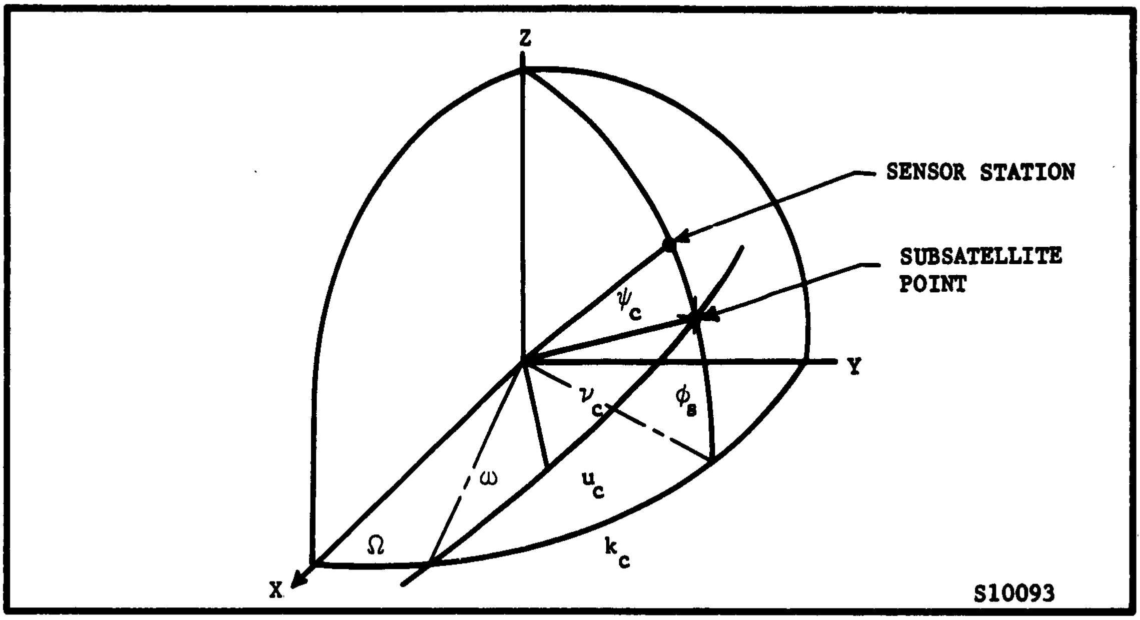


FIGURE 3-1. SUBSATELLITE TRACK

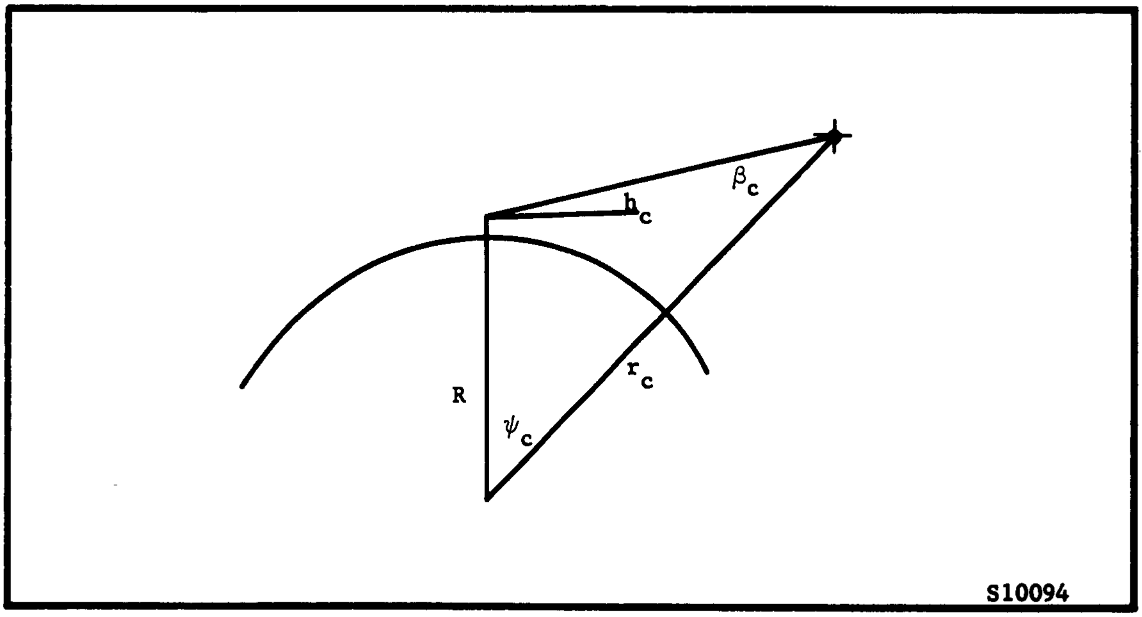


FIGURE 3-2. MERIDIAN OF SENSOR

$$\beta_c = \sin^{-1}\left[\frac{R \cos h_c}{r_c}\right]. \tag{3.1.33}$$

The angle u_c and k_c shown in Figure 3-1 are computed by means of

$$u_c = \sin^{-1}\left[\frac{\sin \phi_s}{\sin i}\right] \tag{3.1.34}$$

and

$$k_c = \sin^{-1}\left[\frac{\tan \phi_s}{\tan i}\right]. \tag{3.1.35}$$

The quadrants of u_c and k_c are determined by the direction of the pass over the sensor and the signs of the bracketed quantities. The angles Ω and ω are obtained from

$$\begin{aligned} \Omega &= \lambda_E - k_c \quad \text{if } i \leq \pi/2 \\ &= \lambda_E + k_c \quad \text{if } i > \pi/2 \end{aligned} \tag{3.1.36}$$

and

$$\omega = u_c - v_c. \tag{3.1.37}$$

The eccentric anomaly at culmination is obtained from

$$E_c = \tan^{-1}\left[\frac{\sin E_c}{\cos E_c}\right] \tag{3.1.38}$$

where

$$\sin E_c = \frac{\sqrt{1 - e^2} \sin v_c}{1 + e \cos v_c} \tag{3.1.39}$$

and

$$\cos E_c = \frac{\cos v_c + e}{1 + e \cos v_c}. \tag{3.1.40}$$

The orbit epoch is defined at the satellite culmination over the sensor, meridian and

$$t_o = 0 \qquad \theta_{t_o} = 0 \tag{3.1.41}$$

$$M_o = E_c - e \sin E_c \tag{3.1.42}$$

The computation then continues, as given by ETYPA, to obtain the desired $\underline{M}$ $\underline{N}$ form.

The fourth option on the element input is converted by ETYPD. This input type defines the elements at a particular range, ρ_o, and elevation angle, h_o, from a specified sensor. Along with these quantities, the orbital elements a, e, i, Ω, and ω are also required as input. Initially, the program tests ρ_o to ensure its validity with respect to the orbit elements. This test is

$$\rho_{min} \leq \rho_o \leq \rho_{max}, \tag{3.1.43}$$

where

$$\rho_{min} = q - R \tag{3.1.44}$$

and

$$\rho_{max} = \sqrt{q_2^2 - R^2} \tag{3.1.45}$$

Perigee distance, q, and apogee distance, q_2, in the above equations are obtained from

$$q = a(1 - e) \tag{3.1.46}$$

and

$$q_2 = a(1 + e) \; . \tag{3.1.47}$$

If ρ_o is valid, the routine goes on to compute the geocentric distance to the satellite, r_o,

$$r_o = \sqrt{R^2 + \rho_o^2 + 2R\,\rho_o \sin h_o} \tag{3.1.48}$$

and the true anomaly

$$v_o = \cos^{-1}\left[\frac{p - r_o}{e\, r_o}\right], \tag{3.1.49}$$

where p is computed from Equation (3.1.5), and v_o is double valued.

The components of the unit vectors $\underline{P}$ and $\underline{Q}$ in the direction of motion in the orbit plane are computed from

$$\begin{aligned} P_x &= \cos\omega \cos\Omega - \sin\omega \sin\Omega \cos i \\ P_y &= \cos\omega \sin\Omega + \sin\omega \cos\Omega \cos i \\ P_z &= \sin\omega \sin i \end{aligned} \tag{3.1.50}$$

and

$$\begin{aligned} Q_x &= -\sin\omega \cos\Omega - \cos\omega \sin\Omega \cos i \\ Q_y &= -\sin\omega \sin\Omega + \cos\omega \cos\Omega \cos i \\ Q_z &= \cos\omega \sin i . \end{aligned} \tag{3.1.51}$$

The components of two $\underline{U}$ vectors are now obtained from the two solutions to v_o by

$$\underline{U}_i = \cos v_{o_i}\, \underline{P} + \sin v_{o_i}\, \underline{Q}, \quad (i = 1,2). \tag{3.1.52}$$

Auxiliary angles λ', which are the local sidereal times of the station for the solutions of the $\underline{U}$ vectors, are established by

$$\lambda_i' = \tan^{-1}\left[\frac{U_y}{U_x}\right]_i \pm 2\tan^{-1}\sqrt{K_i}, \quad (i = 1,2) \tag{3.1.53}$$

where

$$K_i = \frac{\sqrt{1 - U_{z_i}^2}\cos\Phi + U_{z_i}\sin\Phi - \cos\epsilon}{\sqrt{1 - U_{z_i}^2}\cos\Phi - U_{z_i}\sin\Phi + \cos\epsilon}, \quad (i = 1,2) \tag{3.1.54}$$

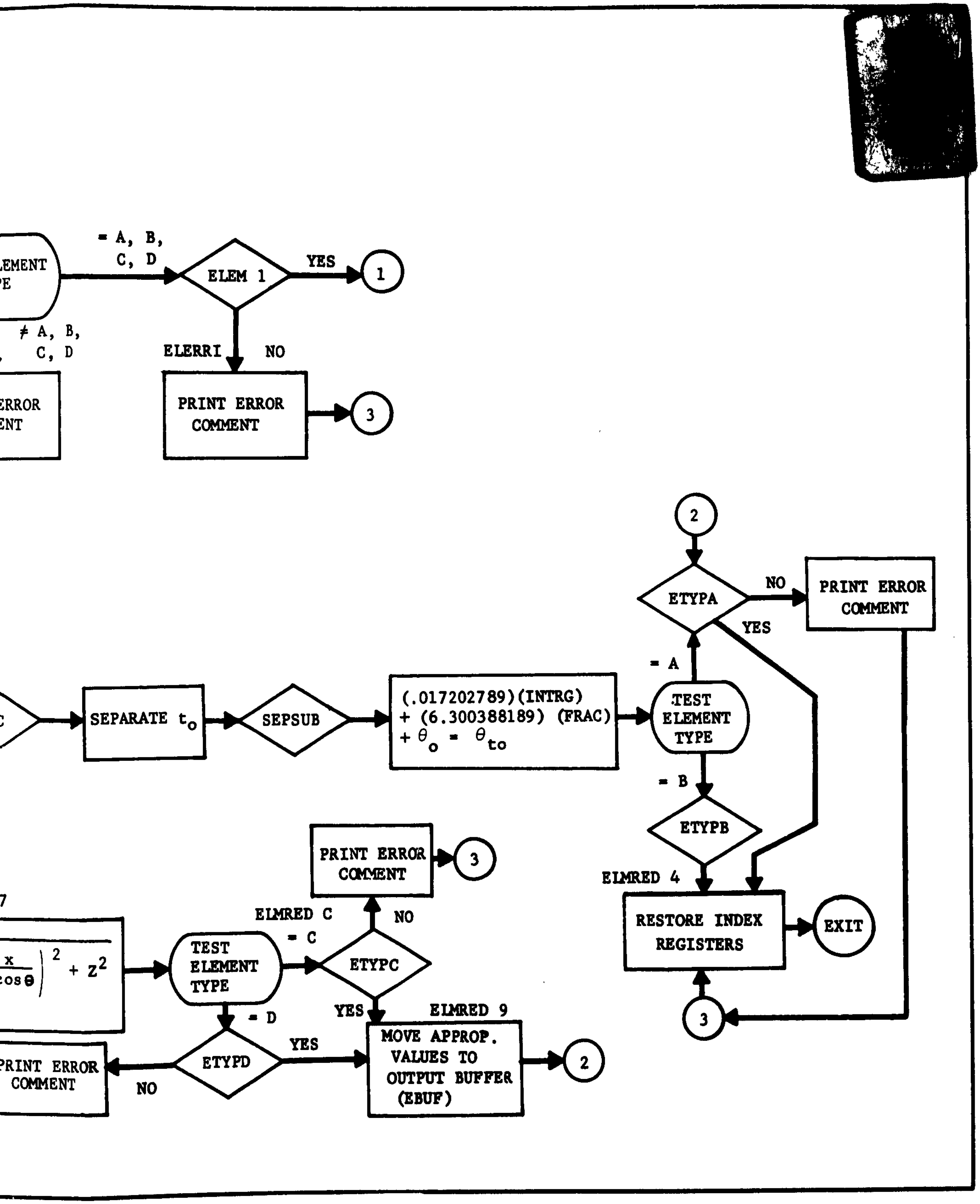


FIGURE 3-3. ELMRED

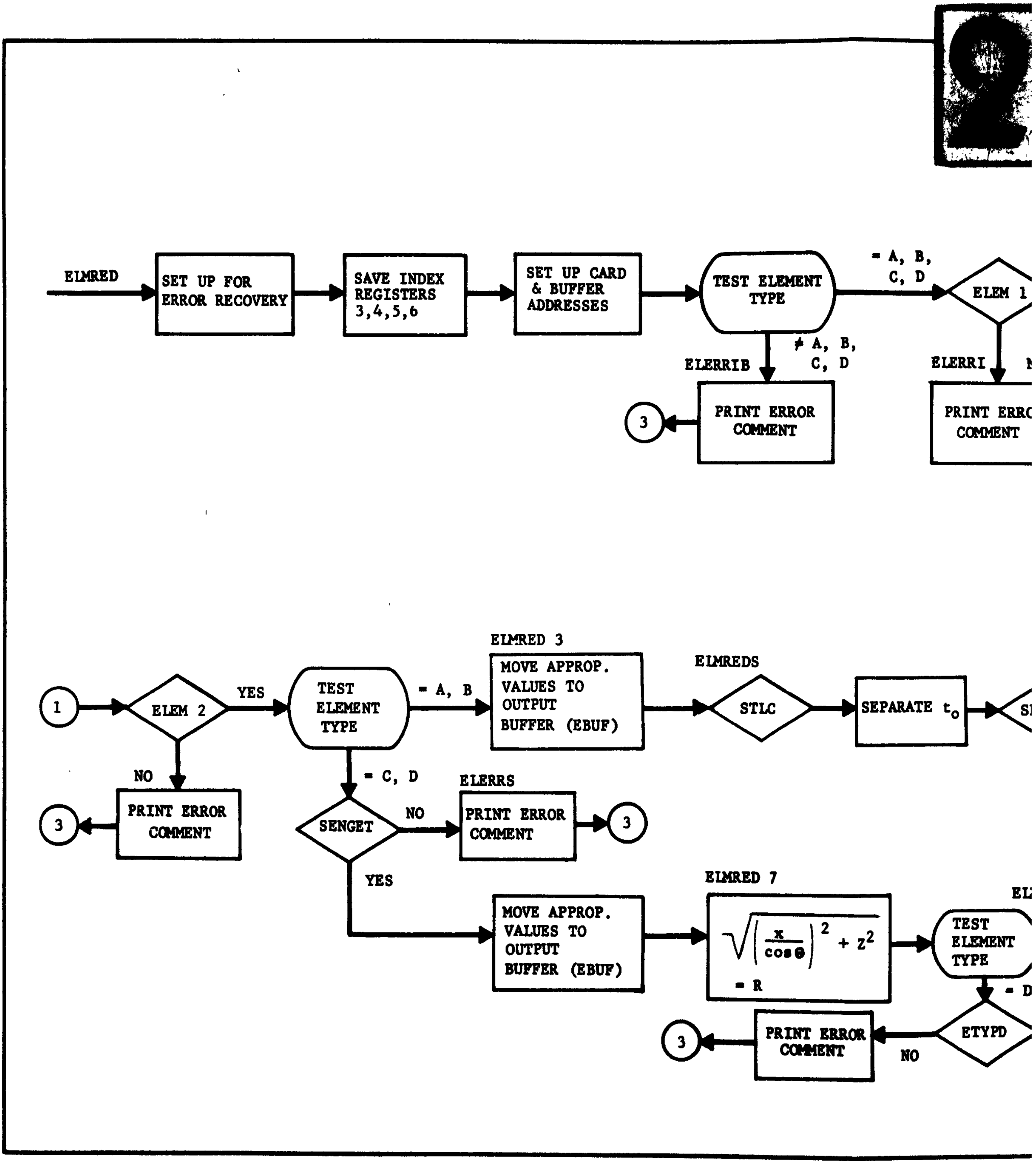

ELMRED
SET UP FOR ERROR RECOVERY
SAVE INDEX REGISTERS 3,4,5,6
SET UP CARD & BUFFER ADDRESSES
TEST ELEMENT TYPE
= A, B, C, D
ELEM 1
≠ A, B, C, D
ELERRIB
PRINT ERROR COMMENT
3
ELERRI
PRINT ERRC COMMENT
1
ELEM 2
YES
NO
PRINT ERROR COMMENT
3
TEST ELEMENT TYPE
= A, B
ELMRED 3
MOVE APPROP. VALUES TO OUTPUT BUFFER (EBUF)
ELMREDS
STLC
SEPARATE t_o
= C, D
SENGET
NO
ELERRS
PRINT ERROR COMMENT
3
YES
MOVE APPROP. VALUES TO OUTPUT BUFFER (EBUF)
ELMRED 7
$\sqrt{\left(\frac{x}{\cos\theta}\right)^2 + z^2} = R$
EL
TEST ELEMENT TYPE
= D
ETYPD
NO
PRINT ERROR COMMENT
3

FIGURE 3-3.

$$\cos \epsilon = \frac{R}{r_o - \rho_o \sin h_o} \qquad (3.1.55)$$

Of the two pairs of possible solutions, one pair is arbitrarily selected and the λ_i' in which the satellite is entering the field of the sensor is chosen as epoch rather than the one in which it is exiting the field.

The Greenwich sidereal time at epoch for either of the solutions is obtained from

$$\theta_{t_o} = \lambda_i' - \lambda_E \qquad (3.1.56)$$

The eccentric anomaly, E_o, at satellite epoch, θ_{t_o}, is computed by Equations (3.1.38), (3.1.39), and (3.1.40) (with subscripts c replaced by o). The mean anomaly M_o is then obtained by

$$M_o = E_o - e \sin E_o. \qquad (3.1.57)$$

3.1.3 SCHEMATIC COMPUTATIONAL SEQUENCE

As previously mentioned, the programs used in the initializing process are executive subroutines. The major routines ELMRED and SENCARD and their associated routines are described fully in Section 5. The flow diagram of ELMRED is shown here (Figure 3-3) for continuity, since it performs the major initializing function and indicates the role of SENCARD and the associated routines.

3.2 ACQUISITION TIME INTERVALS - AQTINT

The time of satellite acquisition for observation from a given station and sensor is governed by two main considerations. The first consideration is the orbit-plane orientation requirement for sensor visibility, and, secondly, it is desirable to specify the acquisition times in such a manner as to provide flexibility in the sensor's observing schedule. During the first and following passes of the satellite, consistent with an observation duration of selected dates, a specification of the initial and final acquisition times which corresponds to convenient sensor initial and final elevation angles will provide a wide range of observation times. The subdivision of the interval between the initial and final times into observation segments yields several initial and final observation times available for selection according to a desired observation schedule.

3.2.1 AQTINT INPUT-OUTPUT DATA

In the formulation of acquisition times in accordance with the above considerations, it is assumed the following quantities are available as input data:

(1) EBLOC, the M N satellite orbit elements and sidereal time at satellite epoch.

(2) SBLOC, sensor identification and sensor station coordinates.

(3) IBLOC, the initial and final elevation angles convenient for the sensor, total number of observation segments per satellite pass, the segments chosen for observations, and observation time frequency.

(4) Time duration of observation.

The output from the program is a block of data, TBLOC, consisting of the times of acquisition from each sensor. The individual times depend on the simulation parameters, Item (3), and the number of acquisition times depends on the observation duration; however, the total number of times is limited to approximately 600. An optional feature of the program enables this block of acquisition times to be printed along with each corresponding sensor identification.

3.2.2 AQTINT FORMULATION

The task of finding the observation times for a given satellite orbit and station is first concerned with the problem of determining the conditions necessary for the satellite to be above the observer's horizon. The examination of the geometric relationships between the orbit or orientation and position of the observer's station yield the following conditions:

(1) If $\eta > 1$ (3.2.1)

and $\nu_1 > \nu$

a portion of the satellite orbit track will always be above the observer's horizon.

(2) If $\eta_1 > 1$ (3.2.2)

and $\nu_1 < \nu$

the satellite will never be visible from the station.

(3) If $\eta_1 < 1$ or $\eta < 1$ (3.2.3)

a portion of the satellite orbit path will be above the observer's horizon only when

$$\nu_1 > \nu \tag{3.2.4}$$

The quantities in Equations (3.2.1) through (3.2.4) are defined by the expressions:

$$\nu = \left[\frac{1}{a(1 + e)}\right]^2 = \left[\frac{1 - \sqrt{a_{xn}^2 + a_{yn}^2}}{p}\right]^2 \tag{3.2.5}$$

$$\nu_1 = 1 - \frac{(h_x X + h_y Y + h_z Z)^2}{(X^2 + Y^2 + Z^2) \quad p} \tag{3.2.6}$$

$$\eta_1 = \frac{\frac{h_z}{\sqrt{p}} \sin \phi - \sqrt{1 - \nu}}{\cos \phi \sqrt{1 - (h_z)^2/p}} \tag{3.2.7}$$

$$\eta = \frac{\frac{h_z}{\sqrt{p}} \sin\phi - \sqrt{1 - \nu_2}}{\cos\phi \sqrt{1 - h_z^2 / p}} \tag{3.2.8}$$

$$\nu_2 = \left[\frac{1 + \sqrt{a_{xn}^2 + a_{yn}^2}}{p} \right]^2 \tag{3.2.9}$$

$$X = -x_c \cos\theta \tag{3.2.10}$$

$$Y = -x_c \sin\theta$$

$$Z = -y_c$$

The local sidereal time, θ, is obtained from

$$\theta = \theta_{t_o} + \lambda_E + (0.0043752691)\ t, \tag{3.2.11}$$

where θ_{t_o} is the sidereal time at epoch from EBLOC, λ_E is the east longitude of the sensor, and t is time in minutes relative to satellite epoch. If the condition specified by Equation (3.2.3) prevails, then an iterative procedure must be carried out, advancing time, t, until the condition specified by Equation (3.2.4) is fulfilled. The existence of the condition specified by Equation (3.2.4) is guaranteed by the existence of condition (3).

The satellite will be above the horizon upon satisfaction of the inequalities (3.2.1) or (3.2.4) and when

$$L_{z_h} > 0, \tag{3.2.12}$$

where

$$L_{z_h} = \frac{1}{\rho} (\rho_x \cos\theta \cos\phi + \rho_y \sin\theta \cos\phi + \rho_z \sin\phi) \tag{3.2.13}$$

and

$$\rho = \sqrt{\rho_x^2 + \rho_y^2 + \rho_z^2} \tag{3.2.14}$$

The components of $\underline{\rho}$ are obtained from

$$\rho_x = X + x \tag{3.2.15}$$

$$\rho_y = Y + y$$

$$\rho_z = Z + z$$

where x, y, and z are obtained from the MNREP subroutine. An iterative procedure must be performed in the time variable, t, until the condition specified by Equation (3.2.12) is fulfilled.

The time, t, corresponding to a given satellite elevation, h_{REQ}, is determined by the convergence of the following iterative process. A linear approximation to the time interval from the current time point, obtained through condition (3.2.12), to that for h_{REQ} is

$$\Delta t = \frac{(h_{REQ} - h)\rho}{k_e\,(\underline{\dot{\rho}} \cdot \underline{\mathcal{D}} + \rho\dot{\theta}\mathcal{A}_z)} \tag{3.2.16}$$

giving the next representation point by means of

$$t_{NEXT} = t + \Delta t \tag{3.2.17}$$

The equatorial system components of $\tilde{\underline{D}}$ in Equation (3.2.16) are obtained from the horizon system components by the rotation

$$\tilde{\underline{D}} \begin{cases} \tilde{D}_x = \tilde{D}_{x_h} S_x + \tilde{D}_{y_h} E_x + \tilde{D}_{z_h} Z_x \\ \tilde{D}_y = \tilde{D}_{x_h} S_y + \tilde{D}_{y_h} E_y + \tilde{D}_{z_h} Z_y \\ \tilde{D}_z = \tilde{D}_{x_h} S_z + \tilde{D}_{y_h} E_z + \tilde{D}_{z_h} Z_z \ , \end{cases} \qquad (3.2.18)$$

where

$$\tilde{\underline{D}}_h \begin{cases} \tilde{D}_{x_h} = \sin h \cos A \\ \tilde{D}_{y_h} = - \sin h \sin A \\ \tilde{D}_{z_h} = \cos h \end{cases} \qquad (3.2.19)$$

$$\underline{S} \begin{cases} S_x = \sin \phi \cos \theta \\ S_y = \sin \phi \sin \theta \\ S_z = - \cos \phi \end{cases} \qquad (3.2.20)$$

$$\underline{E} \begin{cases} E_x = - \sin \theta \\ E_y = \cos \theta \\ E_z = 0 \end{cases} \qquad (3.2.21)$$

and

$$\underline{Z} \begin{cases} Z_x = \cos\phi \cos\theta \\ Z_y = \cos\phi \sin\theta \\ Z_z = \sin\phi \end{cases} \tag{3.2.22}$$

The components of the vector $\dot{\underline{\rho}}$ are obtained from

$$\begin{aligned} \dot{\rho}_x &= -Y \dot{\theta} + \dot{x} \\ \dot{\rho}_y &= X \dot{\theta} + \dot{y} \\ \dot{\rho}_z &= \dot{z} , \end{aligned} \tag{3.2.23}$$

where $\dot{x}$, $\dot{y}$, and $\dot{z}$ are available from MNREP, and θ is the rotation rate of the earth. The azimuth, A, and the elevation angle, h, in Equation (3.2.19) are computed from the components of the $\underline{L}_h$ vector. These expressions are

$$h = \tan^{-1} \left[\frac{L_{z_h}}{\sqrt{1 - L_{z_h}^2}} \right] , \quad 0 \leq h \leq 90^\circ \tag{3.2.24}$$

and

$$A = \tan^{-1} \left[- \frac{L_{y_h}}{L_{x_h}} \right] , \quad 0 \leq A \leq 360^\circ \tag{3.2.25}$$

where

$$L_{x_h} = \frac{1}{\rho}\left[\rho_x \cos\theta \sin\phi + \rho_y \sin\phi \sin\theta - \rho_z \cos\phi\right] \quad (3.2.26)$$

$$L_{y_h} = \frac{1}{\rho}(\rho_y \cos\theta - \rho_x \sin\theta) , \quad (3.2.27)$$

and L_{z_h} is obtained from Equation (3.2.13).

Finally, the initial and final observation times corresponding to the i^{th} observation segment are obtained from the expressions

$$t_I = T_I + \frac{(SEGNOS - 1)}{NUSEG} (T_F - T_I) \quad (3.2.28)$$

and

$$t_F = T_I + \frac{(SEGNOS) + n}{NUSEG} (T_F - T_I) \quad (3.2.29)$$

where

T_I = time in minutes relative to satellite epoch corresponding to the initial elevation angle, h_I.

T_F = time in minutes relative to satellite epoch corresponding to the final elevation angle, h_F.

n = number of consecutive segments minus 1.

NUSEG = total number of observation segments in one satellite pass.

SEGNOS = number, identifying the i^{th} selection observation segment.

The latter two quantities are inputs, where NUSEG may be any number of segments up to a maximum of nine. SEGNOS is derived from input specifying any combination of the segments defined by NUSEG as is the integer, n.

3.2.3 AQTINT PROGRAM SPECIFICATIONS SUMMARY

a. System Input

SBLOC
IBLOC
TOPT
OBSDUR
PERT

b. Associated Input Cards

E CARD
S CARD
AQTINT CONTROL
I CARD

c. System Output

TBLOC

d. System Common Subroutines

ARCTAN
EUNPCK
EXEND
GLOP
IGET
PANT
ARCOS
INITI
MNREP
NXJOB
PAGECON
SENGET
SEPSUB
ARCSIN
SETJC
HEAD
SFXINT
SFXFLT
CARDER

e. Philco Subroutines

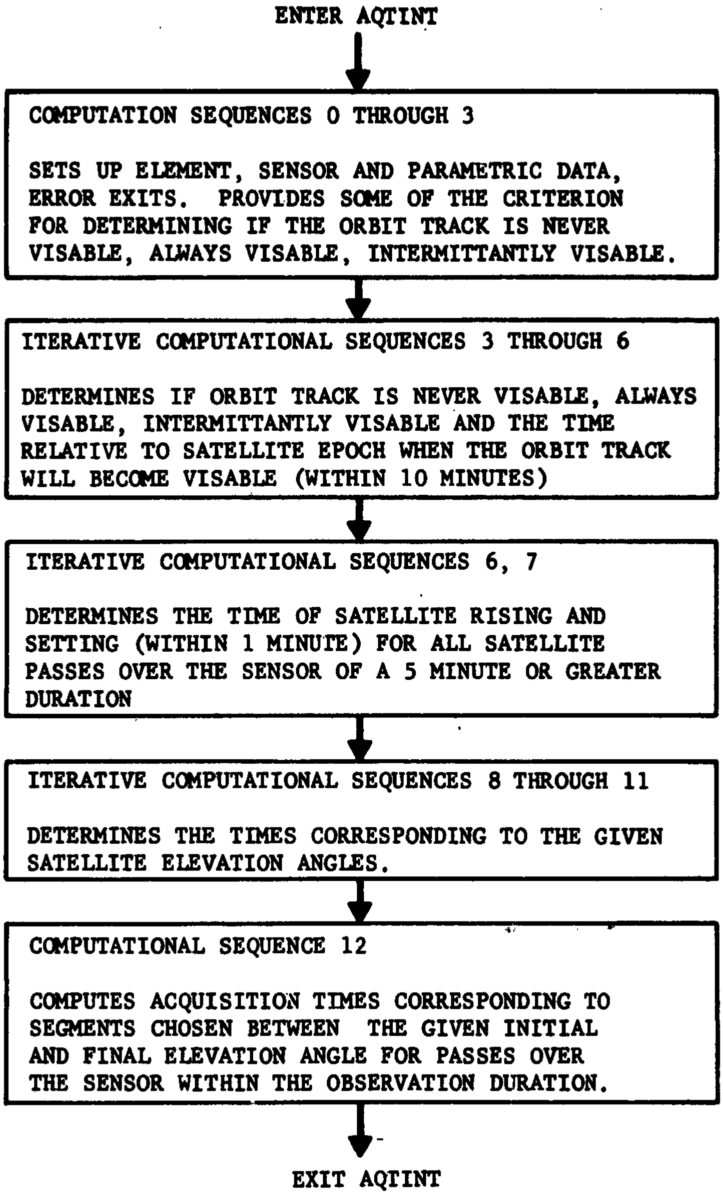
FSIN
FCOS
FSQRT
FACOS
FASIN

3.2.4 AQTINT SCHEMATIC COMPUTATIONAL SEQUENCE

The following glossary provides the definitions of abbreviations used in the AQTINT flow diagrams. These diagrams include an overall description of the program logic (Figure 3-4) followed by a detailed description of the program logic (Figures 3-5, 3-6, 3-7).

Location	Quantity
IDELTAT	Time frequency of observation
ROMAX	ρ_{max} = sensor maximum range of observation
CAPTR	T_R = time of satellite rising
CAPTS	T_S = time of satellite setting
OBSDUR	Observation duration
HI	h_I = given satellite angle at beginning of acquisition interval
HF	h_F = given satellite elevation angle at ending of acquisition interval
DIRI	Indicates rising or setting beginning elevation
DIRF	Indicates rising or setting ending elevation
DTDENOM	Denominator of Δt (Equation 3.2.16)
DELTAT	Δt (Equation 3.2.16)
NU	ν (Equation 3.2.5)
NU_2	ν_2 (Equation 3.2.9)
ETA	η (Equation 3.2.8)
ETA_1	η_1 (Equation 3.2.7)
DELTE	Time step (min.)
DELTP	Time step (min.)
T	Current time since satellite epoch

Location	Quantity
LOOPO	Counters
LOOP	
$LOOP_1$	
$LOOP_2$	
TT	Special T
DIRS	Contains DIRI or DIRF
HS	Contains HI or HF
TTEST	Special T
TTEST O	
AQTHET	Computes sin θ , cos θ (Equation 3.2.11)
AQCAPR	Computes $\underline{R}$ (Equation 3.2.10)
$AQNU_1$	Computes ν_1 (Equation 3.2.6)
AQRHO	Computes $\underline{\rho}$ (Equation 3.2.14)
AQELEV	Computes h (Equation 3.2.24)
AQDELT	Computes Δ t (Equation 3.2.16)
ZAGOUT	Outputs simulation parameter summary
AQOUT	Outputs T Bloc if specified



ENTER AQTINT

COMPUTATION SEQUENCES 0 THROUGH 3

SETS UP ELEMENT, SENSOR AND PARAMETRIC DATA, ERROR EXITS. PROVIDES SOME OF THE CRITERION FOR DETERMINING IF THE ORBIT TRACK IS NEVER VISABLE, ALWAYS VISABLE, INTERMITTANTLY VISABLE.

ITERATIVE COMPUTATIONAL SEQUENCES 3 THROUGH 6

DETERMINES IF ORBIT TRACK IS NEVER VISABLE, ALWAYS VISABLE, INTERMITTANTLY VISABLE AND THE TIME RELATIVE TO SATELLITE EPOCH WHEN THE ORBIT TRACK WILL BECOME VISABLE (WITHIN 10 MINUTES)

ITERATIVE COMPUTATIONAL SEQUENCES 6, 7

DETERMINES THE TIME OF SATELLITE RISING AND SETTING (WITHIN 1 MINUTE) FOR ALL SATELLITE PASSES OVER THE SENSOR OF A 5 MINUTE OR GREATER DURATION

ITERATIVE COMPUTATIONAL SEQUENCES 8 THROUGH 11

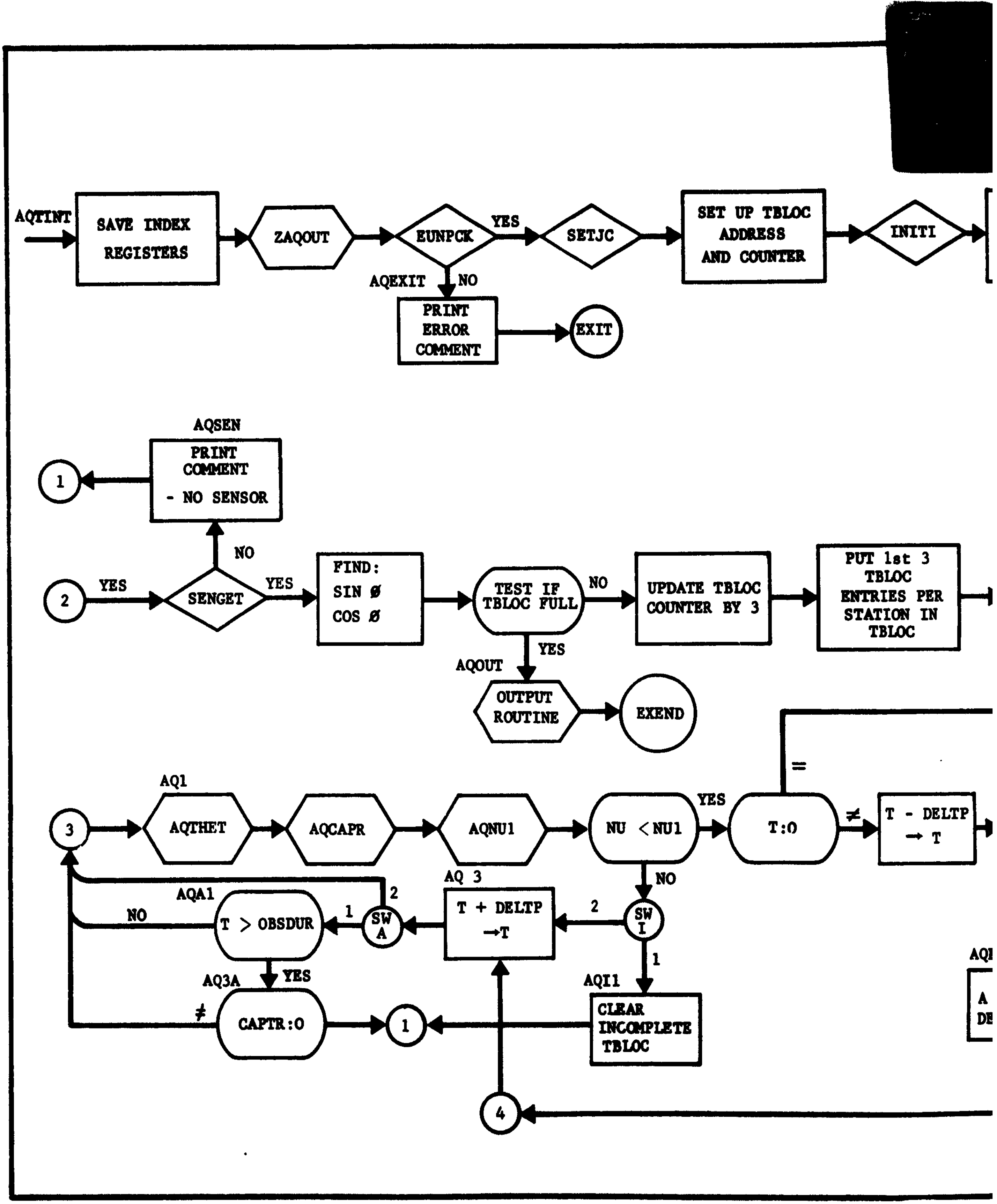
DETERMINES THE TIMES CORRESPONDING TO THE GIVEN SATELLITE ELEVATION ANGLES.

COMPUTATIONAL SEQUENCE 12

COMPUTES ACQUISITION TIMES CORRESPONDING TO SEGMENTS CHOSEN BETWEEN THE GIVEN INITIAL AND FINAL ELEVATION ANGLE FOR PASSES OVER THE SENSOR WITHIN THE OBSERVATION DURATION.

EXIT AQTINT

FIGURE 3-4. AQTINT MACRO FLOW


AQTINT
SAVE INDEX REGISTERS
ZAQOUT
EUNPCK
YES
NO
AQEXIT
SETJC
SET UP TBLOC ADDRESS AND COUNTER
INITI
PRINT ERROR COMMENT
EXIT
AQSEN
PRINT COMMENT - NO SENSOR
1
NO
2
YES
SENGET
YES
FIND: SIN Ø COS Ø
TEST IF TBLOC FULL
NO
UPDATE TBLOC COUNTER BY 3
PUT 1st 3 TBLOC ENTRIES PER STATION IN TBLOC
YES
AQOUT
OUTPUT ROUTINE
EXEND
=
AQ1
3
AQTHET
AQCAPR
AQNU1
NU < NU1
YES
T:0
≠
T - DELTP → T
NO
AQ 3
T + DELTP →T
2
SW I
1
AQA1
NO
T > OBSDUR
1
SW A
2
YES
AQ3A
AQI1
≠
CAPTR:0
1
CLEAR INCOMPLETE TBLOC
4
AQI
A
DE

FI

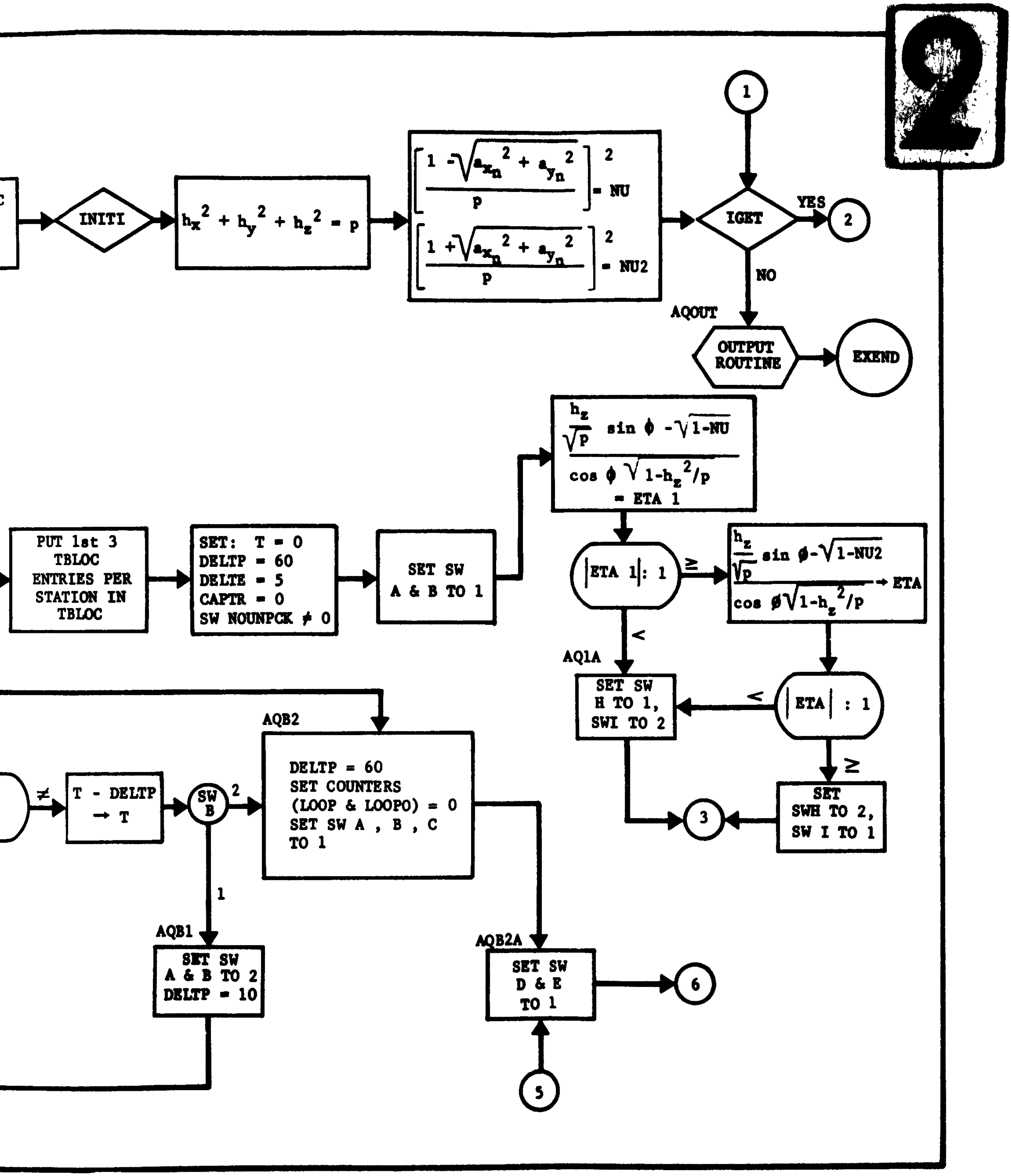
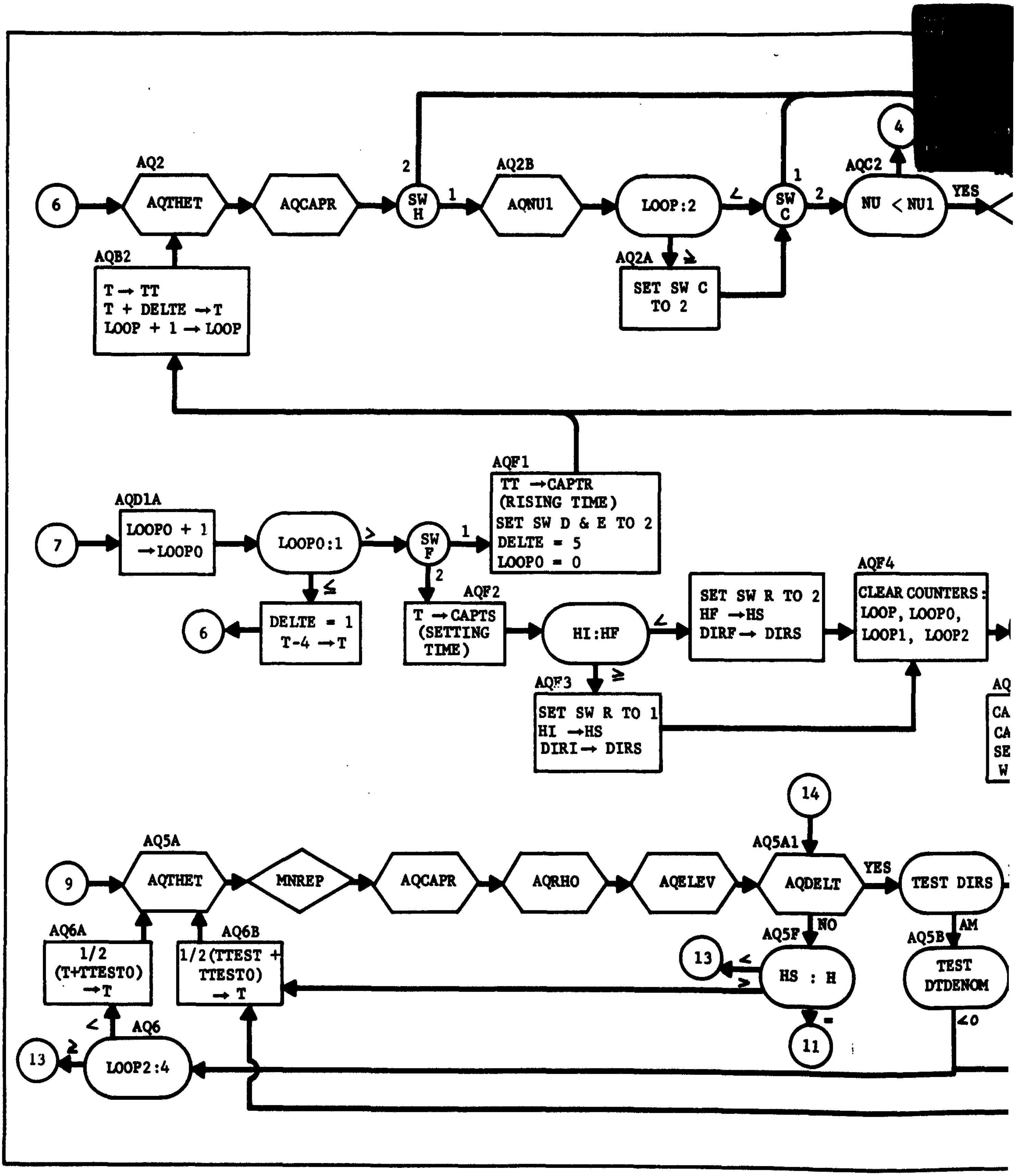

2
INITI
$h_x^2 + h_y^2 + h_z^2 = p$
$\left[\frac{1 - \sqrt{a_{x_n}^2 + a_{y_n}^2}}{p}\right]^2 = NU$
$\left[\frac{1 + \sqrt{a_{x_n}^2 + a_{y_n}^2}}{p}\right]^2 = NU2$
1
IGET
YES
2
NO
AQOUT
OUTPUT ROUTINE
EXEND
$\frac{\frac{h_z}{\sqrt{p}} \sin \phi - \sqrt{1-NU}}{\cos \phi \sqrt{1-h_z^2/p}} = ETA\ 1$
PUT 1st 3 TBLOC ENTRIES PER STATION IN TBLOC
SET: T = 0
DELTP = 60
DELTE = 5
CAPTR = 0
SW NOUNPCK ≠ 0
SET SW A & B TO 1
|ETA 1|: 1
≥
$\frac{\frac{h_z}{\sqrt{p}} \sin \phi - \sqrt{1-NU2}}{\cos \phi \sqrt{1-h_z^2/p}} \rightarrow ETA$
<
AQ1A
SET SW H TO 1, SWI TO 2
<
|ETA| : 1
≥
SET SWH TO 2, SW I TO 1
3
AQB2
≠
T - DELTP → T
SW B
2
DELTP = 60
SET COUNTERS
(LOOP & LOOPO) = 0
SET SW A , B , C
TO 1
1
AQB1
SET SW A & B TO 2
DELTP = 10
AQB2A
SET SW D & E TO 1
6
5

FIGURE 3-5. AQTINT


6
AQ2
AQTHET
AQCAPR
2
SW H
1
AQ2B
AQNU1
LOOP:2
<
AQ2A
≥
SET SW C TO 2
SW C
1
2
4
AQC2
NU < NU1
YES
AQB2
T → TT
T + DELTE → T
LOOP + 1 → LOOP
AQF1
TT → CAPTR (RISING TIME)
SET SW D & E TO 2
DELTE = 5
LOOP0 = 0
7
AQD1A
LOOP0 + 1 → LOOP0
LOOP0:1
>
≤
SW F
1
2
DELTE = 1
T-4 → T
6
AQF2
T → CAPTS (SETTING TIME)
HI:HF
<
≥
SET SW R TO 2
HF → HS
DIRF → DIRS
AQF4
CLEAR COUNTERS:
LOOP, LOOP0,
LOOP1, LOOP2
AQF3
SET SW R TO 1
HI → HS
DIRI → DIRS
AQ
CA
CA
SE
W
14
AQ5A
9
AQTHET
MNREP
AQCAPR
AQRHO
AQELEV
AQ5A1
AQDELT
YES
TEST DIRS
NO
AM
AQ5F
13
<
>
HS : H
=
11
AQ5B
TEST DTDENOM
<0
AQ6A
1/2 (T+TTESTO) → T
AQ6B
1/2 (TTEST + TTESTO) → T
<
AQ6
13
≥
LOOP2:4

FIGURE

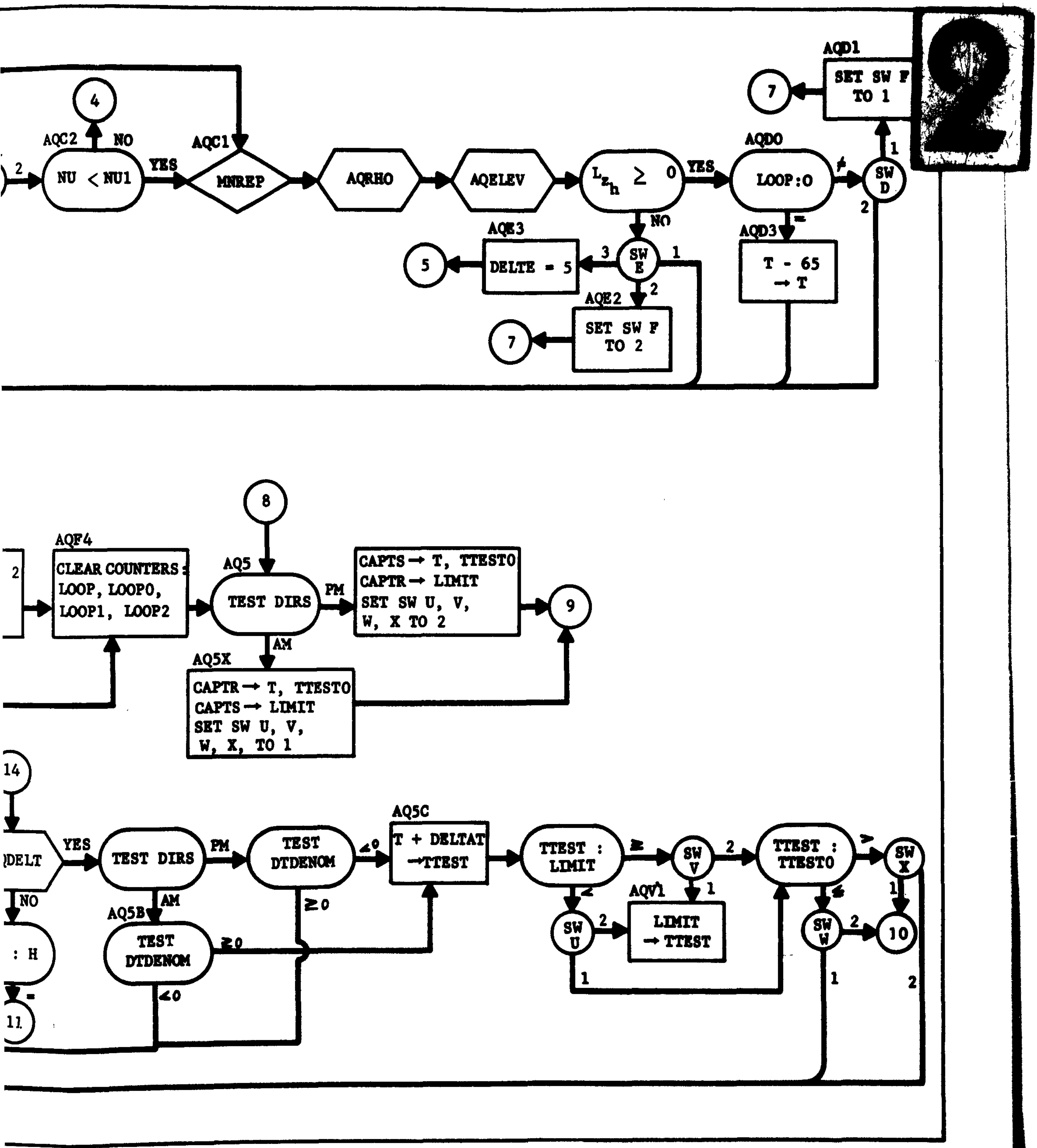
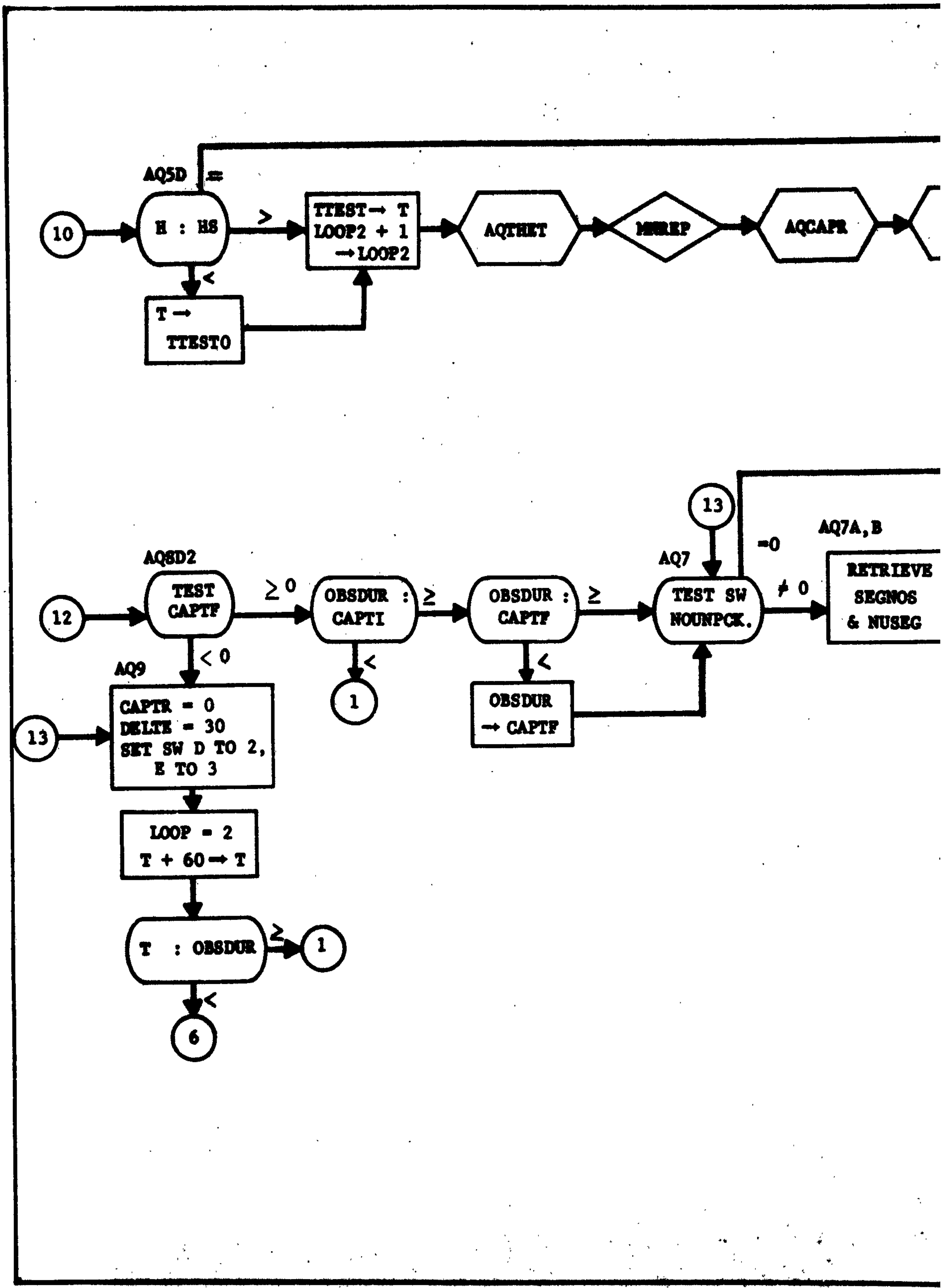


FIGURE 3-6. AQTINT (Continued)


AQ5D
=
10
H : HS
>
<
TTEST → T
LOOP2 + 1
→ LOOP2
T →
TTESTO
AQTHET
MSUREP
AQCAPR
AQSD2
12
TEST
CAPTF
≥ 0
< 0
OBSDUR :
CAPTI
≥
<
1
OBSDUR :
CAPTF
≥
<
OBSDUR
→ CAPTF
13
AQ7
TEST SW
NOUNPCK.
=0
≠ 0
AQ7A,B
RETRIEVE
SEGNOS
& NUSEG
AQ9
13
CAPTR = 0
DELTE = 30
SET SW D TO 2,
E TO 3
LOOP = 2
T + 60 → T
T : OBSDUR
≥
1
<
6

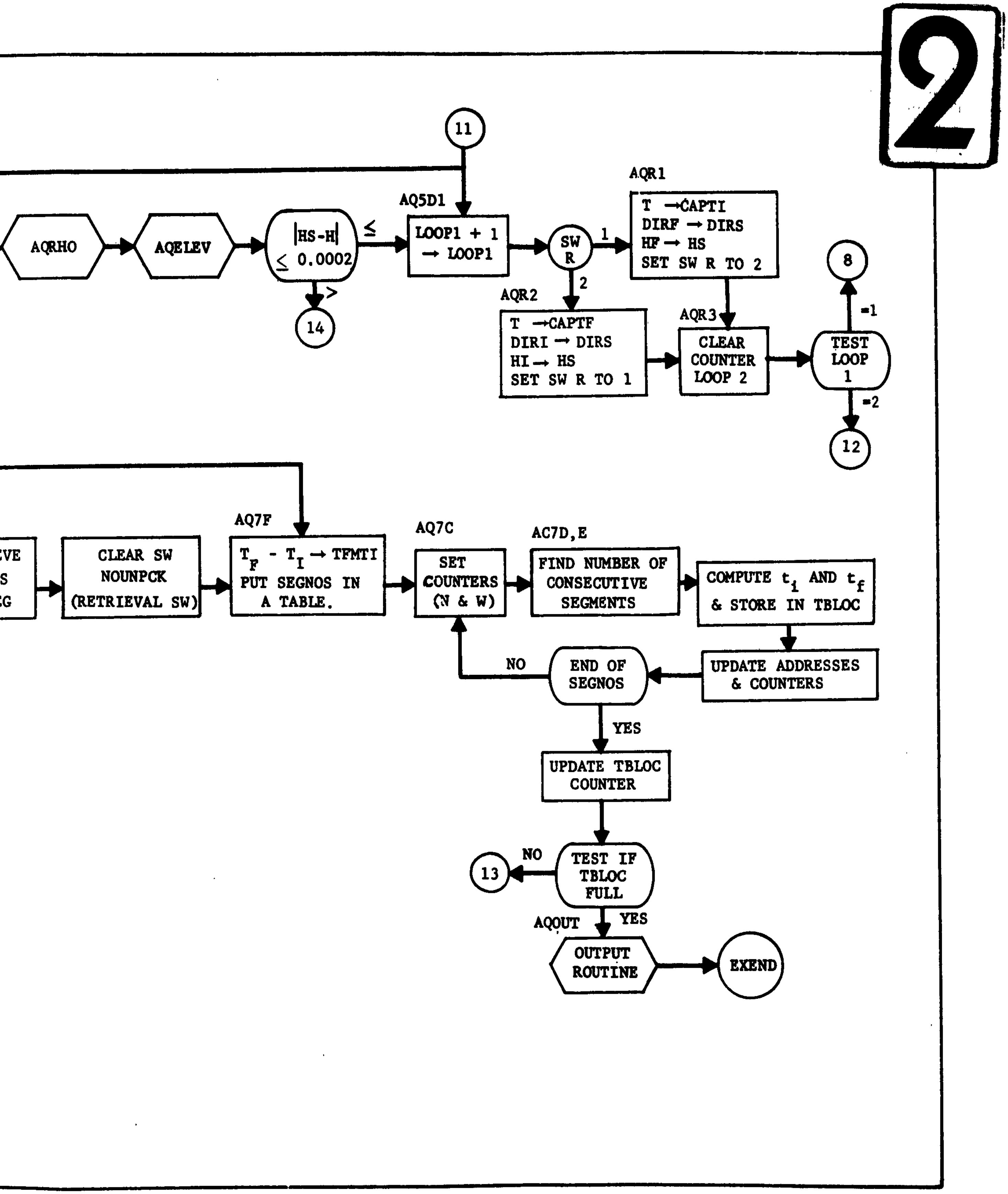


FIGURE 3-7. AQTINT (Continued)

3.3 OBSERVATION SIMULATION - OBSIM

The purpose of the Observation Simulation Program is to compute topocentric coordinates for a satellite pass over a given set of sensors. It is assumed that acquisition times are provided, along with the time frequency of observations desired during the acquisition interval.

3.3.1 OBSIM INPUT-OUTPUT SPECIFICATIONS

The program requires as input the following quantities:

(1) EBLOC, elements of the satellite orbit in $\underline{M}$ $\underline{N}$ representation and sidereal time at epoch

(2) SBLOC, sensor identification and station coordinates of the sensor(s)

(3) TBLOC, initial and final time limits of the acquisition interval for the sensor, sensor observing frequency, sensor observation type, and maximum range of sensor effectiveness.

The output of the program is the block of data, OBLOC, containing the following information:

Sensor number (identification)

Time

Two angular quantities and/or Range and/or
Range rate

The program obtains the simulated observations sensor by sensor. These observations are stored in core to a maximum of 500 points. Any further observations will be punched on observation cards.

3.3.2 OBSIM FORMULATION

The following steps provide the required observation simulation at the points specified by the acquisition interval and the observing frequency for each particular sensor.

The value of the local sidereal time at the required observation time is given by:

$$\theta = (\theta_{t_o} + \lambda_E) + t\ (0.0043752691), \qquad (3.3.1)$$

where λ_E is the east longitude of the station in radians, t is the current time in minutes relative to the satellite epoch, and θ_{t_o} is Greenwich Sidereal time at epoch in radians.

The station position vector components are computed from

$$\underline{R} \begin{cases} X = - x_c \cos\theta \\ Y = - x_c \sin\theta \\ Z = - y_c \end{cases} \qquad (3.3.2)$$

Obtaining the satellite position vector (x, y, z) from MNREP permits the computation of the range vector

$$\underline{\rho} \begin{cases} \rho_x = X + x \\ \rho_y = Y + y \\ \rho_z = Z + z \end{cases} \qquad (3.3.3)$$

and from this, the range to the satellite

$$\rho = \sqrt{\rho_x^{\ 2} + \rho_y^{\ 2} + \rho_z^{\ 2}} \ . \qquad (3.3.4)$$

If the sensor has a limited range, ρ_{max}, then the criterion

$$\rho < \rho_{max}. \qquad (3.3.5)$$

must be satisfied. Also, if the sensor is a visual type, then the following criterion must be fulfilled. The satellite will be visible if illuminated and if the elevation angle of the Sun is negative. The elevation angle of the Sun is negative if, and only if,

$$\sin h_{\odot} = \left[- \frac{X\,L_{x_{\odot}} + Y\,L_{y_{\odot}} + Z\,L_{z_{\odot}}}{\sqrt{(X)^2 + (Y)^2 + (Z)^2}} \right] < 0 \qquad (3.3.6)$$

The satellite will be illuminated if

$$\cos\psi = \frac{1}{r} \left(x L_{x_\odot} + y L_{y_\odot} + z L_{z_\odot}\right) \geq 0 \tag{3.3.7}$$

and if $\cos\psi < 0$, then the satellite will be illuminated if, and only if

$$r \sqrt{1 - \cos^2\psi} > 1 \quad . \tag{3.3.8}$$

The direction cosines to the Sun are computed by

$$\underline{L} \begin{cases} L_{x_\odot} = \cos\delta_\odot \cos\alpha_\odot \\ L_{y_\odot} = \cos\delta_\odot \sin\alpha_\odot \\ L_{z_\odot} = \sin\delta_\odot \end{cases} \tag{3.3.9}$$

where the right ascension of the Sun, $\alpha_\odot$, is obtained from

$$\alpha_\odot = \ell_\odot - C_4 \sin 2\ell_\odot \tag{3.3.10}$$

and the declination, $\delta_\odot$, from

$$\delta_\odot = \tan^{-1} \left[C_5 \sin\alpha_\odot\right] \quad . \tag{3.3.11}$$

Equation (3.3.10) requires the current value of the true longitude of the Sun. This is computed from

$$\ell_\odot = L_\odot + C_1 (t + t_o) + C_2 \left[\sin C_1 (t + t_o) - C_3\right] \quad , \tag{3.3.12}$$

where $L_\odot$, the mean longitude of the Sun at the beginning of the year in radians, and $-C_3$, a corrective constant, are obtained from the STLC subroutine. The other constants in Equations (3.3.10), (3.3.11), and (3.3.12) are

$$C_1 = 0.01720279 \text{ rad/day}$$
$$C_2 = 0.0333358$$
$$C_4 = 0.0431096$$
$$C_5 = 0.4336635$$

The satellite azimuth and elevation angles are computed from

$$A = \tan^{-1}\left[-\frac{L_{y_h}}{L_{x_h}}\right], \quad 0 \leq A \leq 2\pi \tag{3.3.13}$$

and

$$h = \sin^{-1} L_{z_h}, \quad -\frac{\pi}{2} \leq h \leq \frac{\pi}{2} \tag{3.3.14}$$

Equations (3.3.11) and (3.3.12) require the $\underline{L}_h$ vector components

$$\underline{L}_h \begin{cases} L_{x_h} = \frac{1}{\rho}\,(\rho_x \cos\theta \,\sin\phi + \rho_y \sin\theta \sin\phi - \rho_z \cos\phi) \\ L_{y_h} = \frac{1}{\rho}\,(\rho_y \cos\theta - \rho_x \sin\theta) \\ L_{z_h} = \frac{1}{\rho}\,(\rho_x \cos\theta \cos\phi + \rho_y \sin\theta \cos\phi + \rho_z \sin\phi)\,. \end{cases} \tag{3.3.15}$$

The right ascension and declination may be found from

$$\alpha = \tan^{-1}\left[\frac{L_y}{L_x}\right], \quad 0 \leq \alpha \leq 2\pi \tag{3.3.16}$$

$$\delta = \sin^{-1} L_z, -\frac{\pi}{2} \leq \delta \leq \frac{\pi}{2} \tag{3.3.17}$$

Equations (3.3.16) and (3.3.17) require the $\underline{L}$ vector components

$$\underline{L} \begin{cases} L_x = \frac{\rho_x}{\rho} \\ L_y = \frac{\rho_y}{\rho} \\ L_z = \frac{\rho_z}{\rho} \end{cases} \tag{3.3.18}$$

The remaining simulated quantity, range rate, is computed by

$$\dot{\rho} = L_x (\dot{x} + \dot{X}) + L_y (\dot{y} + \dot{Y}) + L_z \dot{z} , \qquad (3.3.19)$$

where $\dot{x}$, $\dot{y}$, and $\dot{z}$ are components of satellite velocity obtained from MNREP, and

$$\dot{X} = -Y\dot{\theta}$$

$$\dot{Y} = X\dot{\theta}$$

$$\dot{\theta} = 0.05883447.$$

3.3.3 OBSIM PROGRAM SPECIFICATIONS SUMMARY

a. System Input

EBLOC
SBLOC
TBLOC
XLSUNO
C3
PERT
OBSOPT

b. Associated Input Cards

E Card
S Card
AQTINT CONTROL

c. System Output

OBLOC

d. System Common Subroutines

ARCTAN
EUNPCK
EXEND
GLOP
IOBPCK
MNREP
NXJOB
OBPCK
PAGECON
PANT
SENGET
SEPSUB
FOBPCK
SETJC
ARCSIN

e. Philco Subroutines

FSIN

FCOS

FATAN

FSQRT

3.3.4 OBSIM SCHEMATIC COMPUTATIONAL SEQUENCE

The Observation Simulation flow diagrams are shown in Figures 3-8 and 3-9. The following glossary of terms explains the references made throughout the diagrams to functions presented in the formulation.

RHO	Computes θ, $\underline{R}$, $\underline{\rho}$, (Equations 3.3.1, 3.3.2, 3.3.3, and 3.3.4).
ANGSUN	Computes $\ell_\odot$, $\alpha_\odot$, $\delta_\odot$, $\underline{L}_\odot$, sin $h_\odot$, $\underline{\mathbf{R}}$ (Equations 3.3.6, 3.3.9, 3.3.10, 3.3.11, and 3.3.12)
LUMANG	Computes cos ψ (Equation 3.3.7).
ELEV	Computes $\underline{L}_h$ (Equation 3.3.15).
AZIM	Computes A (Equation 3.3.13).
RTASCN	Computes $\underline{L}$, α (Equations 3.3.16, 3.3.18).
DECLIN	Computes δ (Equation 3.3.17).
RODT	Computes $\dot{\rho}$ (Equation 3.3.19)
3OUTB OBSOUT	Provides OBSIM output according to input options

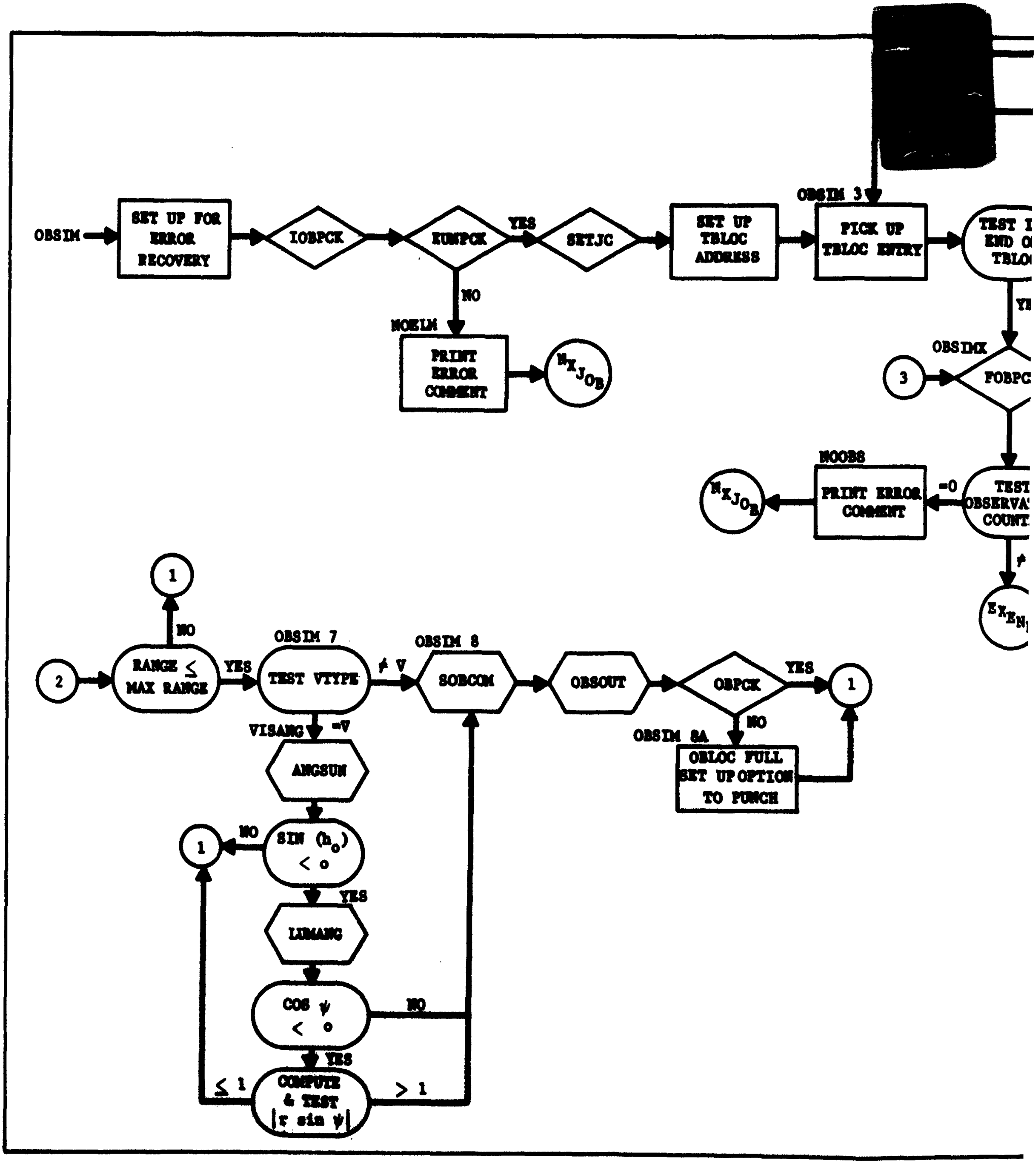

[redacted]
OBSIM
SET UP FOR ERROR RECOVERY
IOBPCK
EUNPCK
YES
SETJC
SET UP TBLOC ADDRESS
OBSIM 3
PICK UP TBLOC ENTRY
TEST I END O TBLO
NO
NOEIM
PRINT ERROR COMMENT
$N_{X_{J_{OB}}}$
YE
OBSIMX
3
FOBPC
NOOBS
=0
TEST OBSERVA COUNT
PRINT ERROR COMMENT
$N_{X_{J_{OB}}}$
≠
$E_{X_{E_{N}}}$
1
NO
OBSIM 7
OBSIM 8
2
RANGE ≤ MAX RANGE
YES
TEST VTYPE
≠ V
SOBCOM
OBSOUT
OBPCK
YES
1
VISANG
=V
NO
OBSIM 8A
ANGSUM
OBLOC FULL SET UP OPTION TO PUNCH
1
NO
SIN (h$_o$) < o
YES
LUMANG
COS ψ < o
NO
YES
≤ 1
COMPUTE & TEST |r sin ψ|
> 1

FIGURE 3-

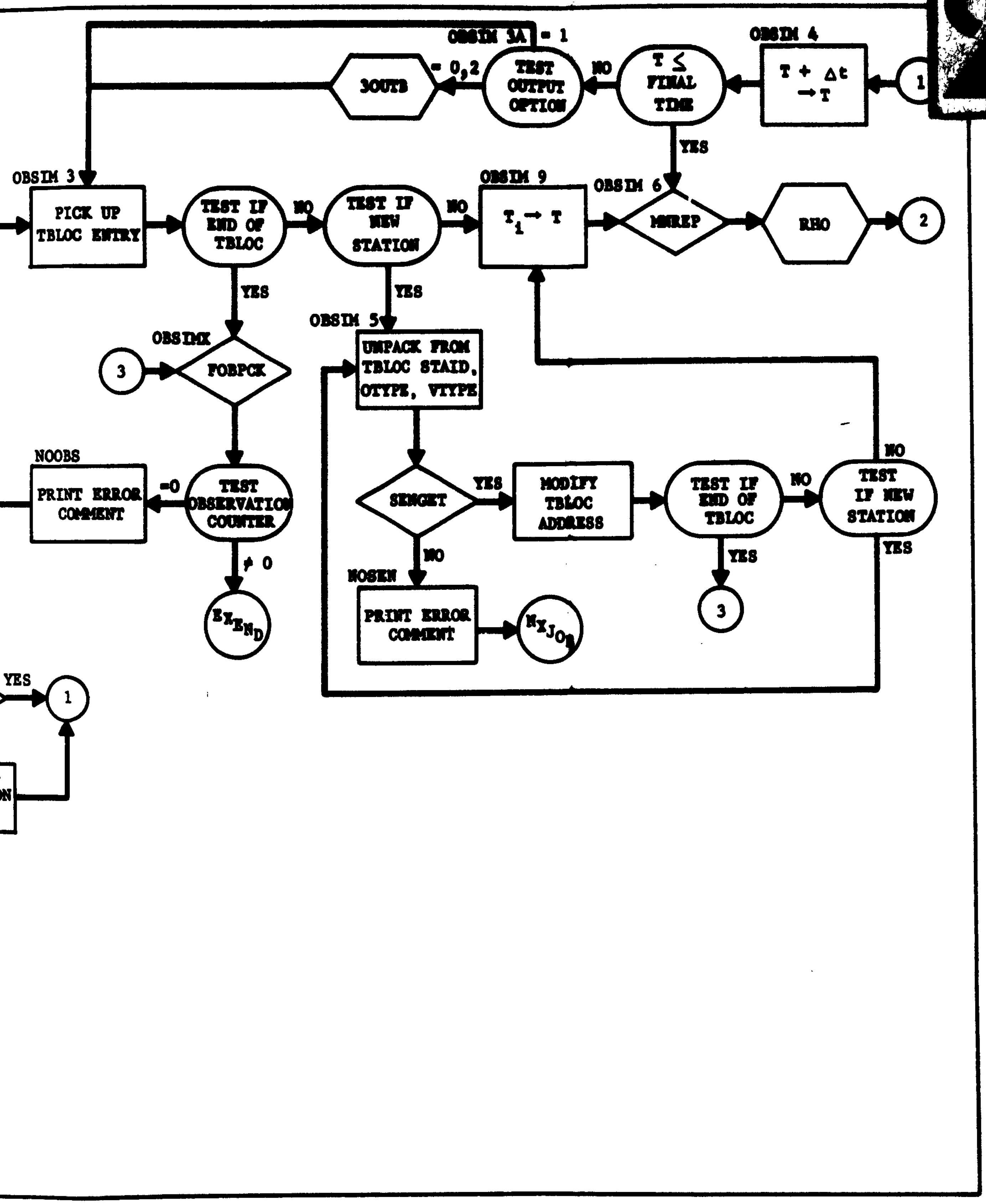


FIGURE 3-8. OBSIM

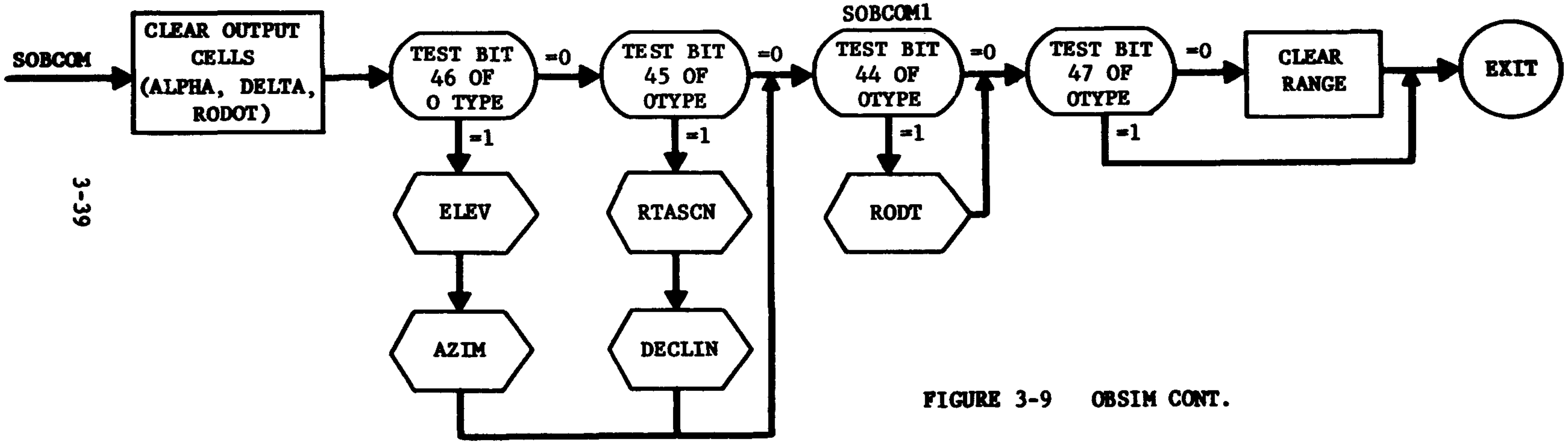


FIGURE 3-9 OBSIM CONT.

3.4 MONTE CARLO - MOCARO

The Monte Carlo, MOCARO, Program Unit is concerned with the generation of sets of contaminated observations from the set of hypothetically true observations. The generation process is based on the hypothesis that the difference between the contaminated and true observations is represented by a random sensor error of fixed standard deviation, σ_{obs}, and a bias error of fixed magnitude. The standard deviation error and bias error are therefore representative of the sensor accuracy.

3.4.1 MOCARO INPUT-OUTPUT DATA

The input data consists of:

(1) OBLOC, true observations

(2) SGBLOC, sensor standard deviation and bias errors

(3) FIRSTRN, first random number used to start the generation of random normal deviates

(4) COFLAG, control option to print out contaminated observations

(5) K, the number of sets of contaminated observation to be generated.

The output is a block of contaminated observations, COBLOC, containing the information:

Sensor number

Time

Two angular quantities and/or

Range and/or

Range rate

3.4.2 MOCARO FORMULATION *

The contaminated observation, OBS_c, is obtained by the relationship,

$$OBS_c = OBS_T + BIAS_{OBS} + X_{RN}\,\sigma_{OBS}$$

where

OBS_T simulated value of the observation from OBLOC

X_{RN} random normal variate

$BIAS_{OBS}$ fixed estimate of bias error of the sensor's measurement

σ_{OBS} fixed estimate of standard deviation of the sensor's measurement

The random normal variates, X_{RN}, are obtained by a generation and subsequent transformation of uniform random variates. The generation of uniform random numbers is obtained from the relationship

$$r_n + 1 = r_n G, \left[\text{MOD } 2^{47}\right] \tag{3.4.1}$$

where G is a large odd number. The starting random number, r_o, is of the form,

$$r_o = 4K + 1 \tag{3.4.2}$$

where K may be any positive integer. Equation (3.4.1) will produce pseudorandom integers between 0 and 2^{47}. The integers are changed to a more convenient fractional form ranging from 0 to 1 by moving the binary point to the first bit position.

*The procedure for generation of the random normal variate is summarized according to the programming development provided by the H. S. Gelman Co. Limited, Toronto, Canada.

The transformation from the uniform random variates to standardized ($\sigma = 1, \mu = 0$) normal variates is governed by the relationship,

$$r = \int_{-\dot{\alpha}}^{X_{RN}} \frac{1}{\sqrt{2\pi}} e^{-(\frac{t^2}{2})} dt. \qquad (3.4.3)$$

Since the closed analytic form of the inverse of this relationship is unknown, a table look-up and interpolation procedure is used to find the normal variates, X_{RN}, from the generated uniform variates, r.

3.4.3 MOCARO PROGRAM SPECIFICATION SUMMARY

a. System Input

OBLOC
SGBLOC
FIRSTRN
MCFIRST
COFLAG
KMAX
CCARD

b. Associated Input Cards

M CARD
MOCARO CONTROL

c. System Output

COBLOC
VELNO

d. System Common Subroutines

COBPCK
GLOP
INITOBS
IGOBPCK
NODN
OBSGET
PAGECON
PANT
REDN
SIGGET
REDNI
EXEND
INITCOB
COBGET
HEAD
INITSIG
SFXINT
CARDER

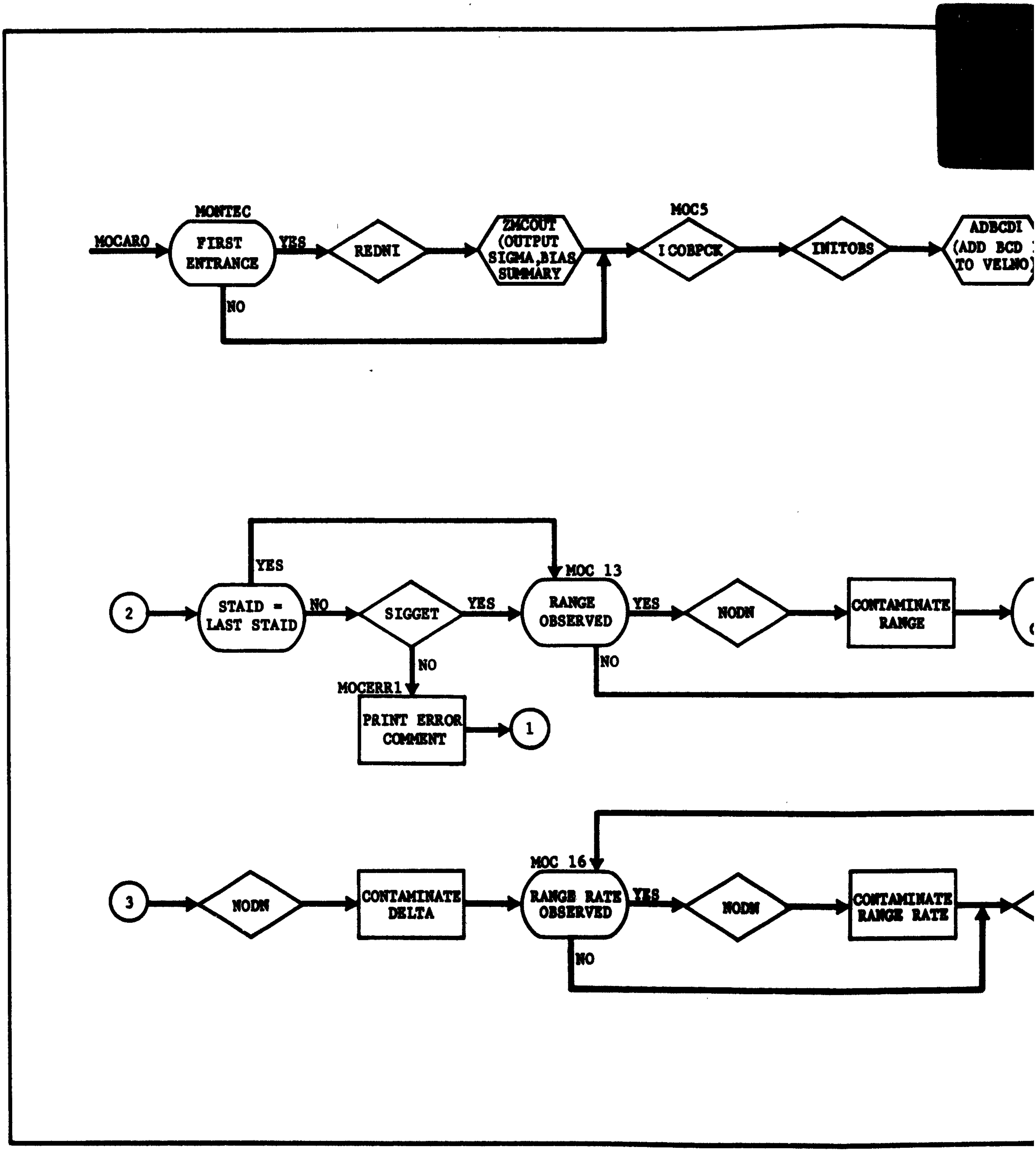

MOCARO
MONTEC
FIRST ENTRANCE
YES
NO
REDNI
ZMCOUT (OUTPUT SIGMA,BIAS SUMMARY
MOC5
ICOBPCK
INITOBS
ADBCDI (ADD BCD TO VELNO)
2
STAID = LAST STAID
YES
NO
SIGGET
YES
NO
MOCERR1
PRINT ERROR COMMENT
1
MOC 13
RANGE OBSERVED
YES
NO
NODN
CONTAMINATE RANGE
3
NODN
CONTAMINATE DELTA
MOC 16
RANGE RATE OBSERVED
YES
NO
NODN
CONTAMINATE RANGE RATE

FIGURE 3-1

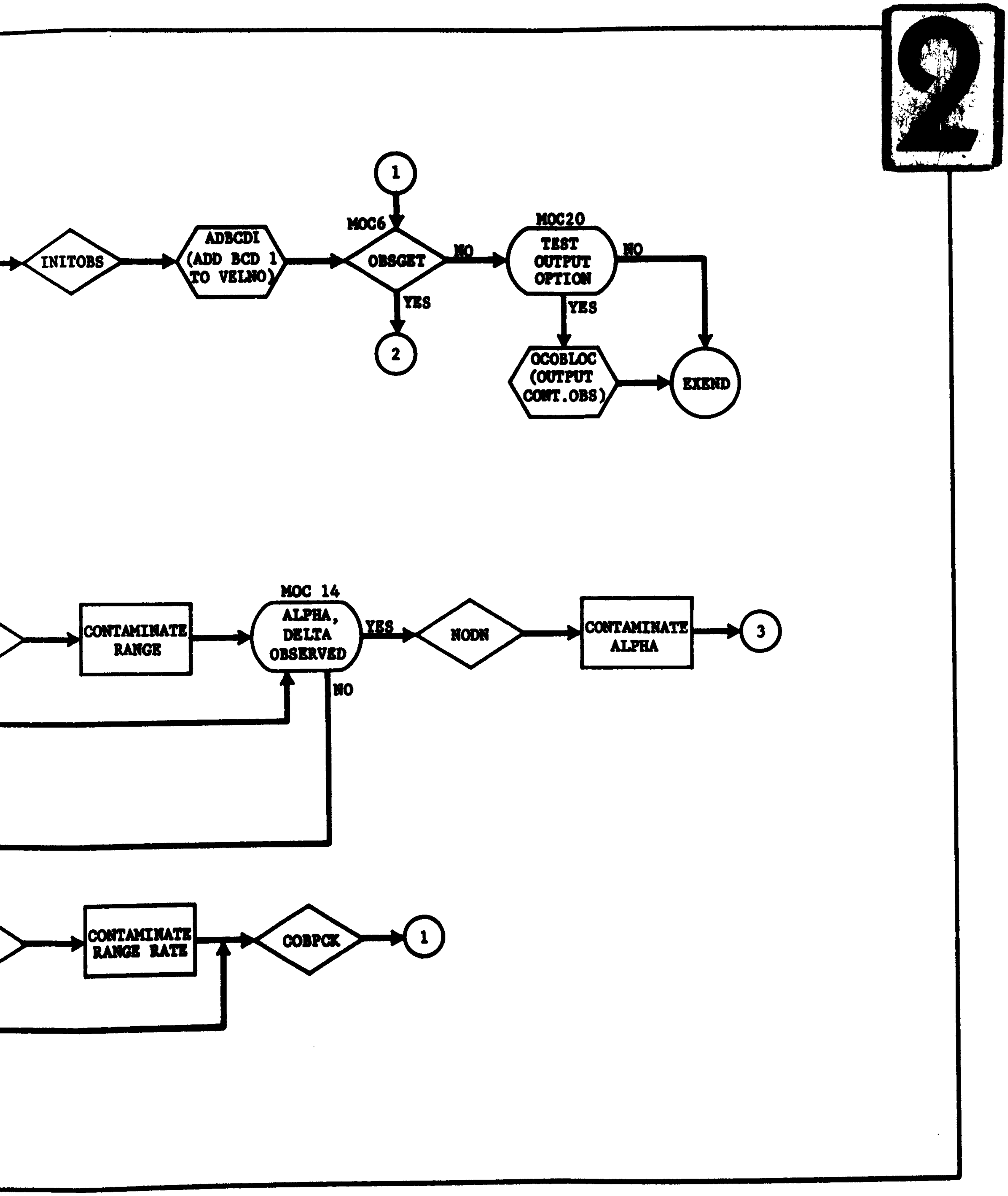


FIGURE 3-10. MOCARO

e. Philco Subroutines

(None)

3.4.4 MOCARO SCHEMATIC COMPUTATIONAL SEQUENCE

The Monte Carlo flow diagrams are presented in Figure 3-10.

3.5 DIFFERENTIAL CORRECTION - SGPDC

The purpose of the differential correction program is to determine the sets of orbital elements which individually best represent, in a least squares sense, the corresponding sets of acceptable input observations. This routine is based on the theory of motion in the two-body problem with the optional provision of including the effects of the secular perturbations on the orbital elements produced by the equatorial bulge of the earth and air drag.

The basic computational sequence in the routine consists of:

(1) the computation of theoretical observations (either with or without perturbations) at the same time points as the input observations on the basis of a preliminary set of input orbital elements.

(2) the formation of observation residuals (observed minus computed).

(3) the comparison of the magnitude of the residuals with maximum values beyond which all residuals are rejected.

(4) formation of linear correction equations containing the residuals and the element corrections (either with or without perturbations).

(5) solution of the least squares matrix of the correction equations for the corrections to the elements.

Following this sequence of computations, the new set of elements are obtained by applying the corresponding corrections to the old set of elements and repeating the above process until the RMS value of the last residuals converge to a desired magnitude considered as acceptable.

The formation of the correction equations is accomplished by use of analytical differential correction formulas. The orbital elements employed in these expressions are:

$\underline{a} = e\,\underline{P}$, a vector with magnitude of eccentricity, e, directed towards perigee

n, the mean motion

$U_o = M_o + \omega$, the mean argument of latitude at arbitrary epoch, t_o, of mean anomaly, M_o, and argument of perigee, ω

Ω = the nodal longitude

i = orbit inclination

c'' = the drag parameter defined by the equation

$$L = L_o + \left(\frac{dL}{dt}\right)_o (t - t_o) + n_o c'' (t - t_o)^2$$

where L is the mean longitude.

The adoption of Ω and i to prescribe the orbit plane orientation and U_o to denote the initial position at epoch, necessarily restricts the application of the differential correction routine to nonequatorial orbits. The conversion of the input element set to the above form is accomplished within the program.

3.5.1 SGPDC INPUT-OUTPUT DATA

a. Numerical Input

The input observational data accepted by the program may be any combination of the following types:

Topocentric right ascension, α, and declination, δ, or azimuth, A, and elevation angle, h.

Slant range, ρ .

Slant range rate, $\dot{\rho}$.

Included with the above observations are the identification number of the observation type, the associated station, and the time of observation. The source of the above observational data is output from the Monte Carlo Error Generator stored in the computer in the contaminated observation block (COBLOC).

The input station coordinates are contained in SBLOC.

The input set of orbital elements are the N M type.

$a_{xN} = \ell \cos \omega$, component of **a** along the nodal line.

$a_{yN} = \ell \sin \omega$, component of **a** perpendicular to the nodal line .

$\underline{h} = \sqrt{p}\ \underline{W}$, the orbital angular momentum vector.

$L_o = M_o + \omega \pm \Omega$, the mean longitude of the satellite at epoch.

C_o = the drag parameter defined by the equation:

$t = t_o + P_{No}\ (N - N_o) + C_o\ (N - N_o)^2$ where N is the revolution number and P_{No} is the nodal period at satellite epoch.

t_o = time of satellite epoch.

This input element set is stored in the element block (EBLOC).

b. Program Control Input Data

(1) Perturbation option.

(2) Number of element correction iterations override.

(3) Specific elements to be corrected override.

(4) Residual root-mean-square factor override.

(5) First absolute maximum range and angle residual override.

(6) Second absolute maximum range and angle residual override.

(7) Absolute maximum range rate residual override.

(8) Residual root-mean-square convergence criterion override.

(9) Option to print out variant classical elements and classical delta elements.

(10) Option to print out variant epoch position and velocity and epoch delta position and velocity.

(11) Option to punch out variant classical elements.

(12) Option to punch out variant epoch position and velocity.

(13) Option to print residual RMS and number of observations used and rejected.

(14) Option to print last iteration residuals.

The above data are inputs to the program via the SGPDC Control card.

c. Output Data

(1) Variant $\underset{\sim}{N}\underset{\sim}{M}$ elements (VEBLOC).

(2) Classical delta elements (DEBLOC).

(3) Optional output data are listed above (Items 9 through 14) under Program Control Data.

3.5.2 SGPDC FORMULATION

The following formulation is intended to serve as description of the fundamental computations in the program and their relation to the program objective, and is not illustrative of the actual computational sequence. The program information flow diagrams (Figures 3-11 through 3-14) describe the details of the complete computational sequence.

a. Preliminary Calculations

This part of the program calculates "observations" of the orbit specified by the input set of elements (or subsequently corrected elements) made from the sensor station. Given the station coordinates of latitude, ϕ , east longitude, λ_e, and height above sea level, H, and Greenwich sidereal time, θ_{to}, at satellite epoch, t_o, the following procedure computes the observations $\dot{\rho}_c$, ρ_c, and the direction cosines of the unit vector, $\underline{L}$, from the station to the satellite for time, t, relative to satellite epoch.

Compute local sidereal time, θ at time, t,

$$\theta = \lambda_e + \theta_{to} + (0.0043752691)t, \text{ (Mod } 2\pi) \tag{3.5.1}$$

Compute the station vector, $\underline{R}$,

$$\left.\begin{aligned} X &= -x_c \cos\theta \\ Y &= -y_c \sin\theta \\ Z &= -y_c \end{aligned}\right\} \qquad (3.5.2)$$

where the x_c, y_c quantities are obtained from (SBLOC) previously stored with the appropriate Station ID.

Compute the slant range, ρ_c,

$$\underline{\rho_c} = \underline{r} + \underline{R}, \qquad (3.5.3)$$

$$\rho_c = \sqrt{\rho_x^2 + \rho_y^2 + \rho_z}^2 \qquad (3.5.4)$$

$$\left.\begin{aligned} \rho_x &= x + X \\ \rho_y &= y + X \\ \rho_z &= z + Z, \end{aligned}\right\} \qquad (3.5.5)$$

where x, y, z are components of $\underline{r}$ obtained from the MNREP subroutine by means of the input elements and the time, t.

Compute unit vector from the station to the satellite in the equatorial coordinate system.

$$\underline{L}_c \left\{ \begin{aligned} L_{xc} &= \frac{\rho_x}{\rho_c} \\ L_{yc} &= \frac{\rho_y}{\rho_c} \\ L_{zc} &= \frac{\rho_z}{\rho_c} \end{aligned} \right. \qquad (3.5.6)$$

Compute range rate, $\dot{\rho}_c$,

$$\underline{\dot{R}}\begin{cases}\dot{X} = -Y\,\dot{\theta}, \dot{\theta} = 0.058,834,47 \\ \dot{Y} = X\dot{\theta} \\ \dot{Z} = 0\end{cases} , \qquad (3.5.7)$$

$$\underline{\dot{\rho}}_c = \underline{\dot{r}} + \underline{\dot{R}} , \qquad (3.5.8)$$

$$\dot{\rho}_c = \underline{L} \cdot \underline{\dot{\rho}}_c = L_x\,(\dot{x} + \dot{X}) + L_y(\dot{y} + \dot{Y}) + L_z\dot{z}, \qquad (3.5.9)$$

where $\dot{x}$, $\dot{y}$, $\dot{z}$ are components of $\underline{\dot{r}}$, obtained from the MNREP subroutine.

b. **Residuals**

The observation residuals are given by the following expressions. The nonsubscripted symbols denote the input observational quantities, and the subscript c denotes the above computed observational quantities.

If range ρ, is observed, the residual is:

$$R_1 = \rho - \rho_c \qquad (3.5.10)$$

If azimuth A, and elevation h, are observed, the residuals are, respectively.

$$R_2 = \rho_c\,\underline{\tilde{A}} \cdot (\underline{L} - \underline{L}_c) \text{ and,} \qquad (3.5.11)$$

$$R_3 = \rho_c\,\underline{\tilde{D}} \cdot (\underline{L} - \underline{L}_c), \qquad (3.5.12)$$

where

$$\underline{\tilde{A}} = \tilde{A}_{xh}\,\underline{S} + \tilde{A}_{yh}\,\underline{E} + \tilde{A}_{zh}\,\underline{Z} , \qquad (3.5.13)$$

$$\underline{\tilde{D}} = \tilde{D}_{xh}\,\underline{S} + \tilde{D}_{yh}\,\underline{E} + \tilde{D}_{zh}\,\underline{Z} \text{ , and} \qquad (3.5.14)$$

$$\underline{L} = L_{xh}\,\underline{S} + L_{yh}\,\underline{E} + L_{zh}\,\underline{Z} . \qquad (3.5.15)$$

The $\underline{S}$, $\underline{E}$, $\underline{Z}$ unit vector system and the horizon oriented $\underline{L}_h$, $\underline{\tilde{A}}_h$, $\underline{\tilde{D}}_h$ unit vector system are defined by:

$$\underline{S}\begin{cases} S_x = \sin\phi\cos\ \theta \\ S_y = \sin\phi\sin\ \theta \\ S_z = -\cos\ \phi, \end{cases} \qquad (3.5.16)$$

$$\underline{E}\begin{cases} E_x = -\sin\theta \\ E_y = \cos\ \theta \\ E_z = 0 \qquad , \end{cases} \qquad (3.5.17)$$

$$\underline{Z}\begin{cases} Z_x = \cos\phi\cos\ \theta \\ Z_y = \cos\phi\sin\ \theta \\ Z_x = \sin\phi \quad , \end{cases} \qquad (3.5.18)$$

$$\underline{L}_h\begin{cases} L_{xh} = -\cos A\cos h \\ L_{yh} = \sin A\cos h \\ L_{zh} = \sin h \quad , \end{cases} \qquad (3.5.19)$$

$$\underline{\tilde{A}}_h\begin{cases} \tilde{A}_{xh} = \sin A \\ \tilde{A}_{yh} = \cos A \\ \tilde{A}_{zh} = 0 \qquad , \text{ and} \end{cases} \qquad (3.5.20)$$

$$\underline{\tilde{D}}_h \begin{cases} \tilde{D}_{xh} = \cos A \sin h \\ \tilde{D}_{yh} = -\sin A \sin h \\ \tilde{D}_{zh} = \cos h. \end{cases} \tag{3.5.21}$$

If right ascension α, and declination δ, are observed, the residuals are:

$$R_4 = \rho_c \underline{A} \cdot (\underline{L} - \underline{L}_c) \tag{3.5.22}$$

$$R_5 = \rho_c \underline{D} \cdot (\underline{L} - \underline{L}_c) \tag{3.5.23}$$

where

$$\underline{A} \begin{cases} A_x = -\sin \alpha \\ A_y = \cos \alpha \\ A_z = 0 \quad , \end{cases} \tag{3.5.24}$$

$$\underline{D} \begin{cases} D_x = -\sin \delta \cos \alpha \\ D_y = -\sin \alpha \sin \delta \\ D_z = \cos \delta, \text{ and} \end{cases} \tag{3.4.25}$$

$$\underline{L} \begin{cases} L_x = \cos \delta \cos \alpha \\ L_y = \cos \delta \sin \alpha \\ L_x = \sin \delta. \end{cases} \tag{3.5.26}$$

If range rate $\dot{\rho}$, is observed,

$$R_6 = \rho_c \Delta \dot{\rho} = (\dot{\rho} - \dot{\rho}_c)\, \rho_c \tag{3.5.27}$$

c. Residual Rejection Test

After forming a set of residuals corresponding to a set of observations, the magnitudes of the angle and range residuals are compared with an absolute maximum value of 1000 km and the range rate residual is compared with an absolute maximum value of 0.5 km/sec. All residuals exceeding these absolute limits are rejected from the current iteration. Following the above first residual rejection test, a second similar residual rejection test is performed with a new maximum value equal to 1.5 x (RMS value of the previously accepted residuals). The remaining number of residuals are used in the correction Equation (3.5.28). However, upon completion of the calculation of the new set of corrected elements, the absolute maximum value of the first rejection test is subject to a change from 1000 km to 75 km if the new RMS value of the residuals is less than 50 km.

d. Correction Equations

This part of the program relates the residuals in the observations at time, t, to the corrections to be applied to the initial orbital parameters at epoch, t_o. The procedure allows for the calculation of the improved set of orbital elements and quantities associated with the station coordinates at the observation time. The procedure permits the combination of these quantities to obtain coefficients of linear relationships relating residuals to any combination of corrections, $\frac{\Delta n}{n}$, Δa_{xn}, Δa_{yn}, ΔU, $\Delta\Omega$, Δi, $\Delta c''$. Finally, the corrections to be applied to the elements are determined by solving the (usually over-determined) system of linear correction equations.

The correction equations have the form:

$$R_i = \left(C_{\frac{\Delta n}{n}}\right)_i \cdot \frac{\Delta n_o}{n_o} + \left(C_{\Delta a_{xN}}\right)_i \Delta a_{xN_o} + \left(C_{\Delta a_{yN}}\right)_i \Delta a_{yN_o} \qquad (3.5.28)$$

$$+ \left(C_{\Delta U}\right)_i \Delta U_o + \left(C_{\Delta\Omega}\right)_i \Delta\Omega_o + \left(C_{\Delta i}\right)_i \Delta i + \left(C_{\Delta c''}\right)_i \Delta c'' ,$$

where the form of the coefficients, C_i, depend on the observation type in residual, R_i, and time of observation. These coefficients which have been developed by means of first order partials are functions of the orbit elements and computed observations.

The coefficients are computed by first establishing the R and V coefficients at time t:

$$\left.\begin{aligned}
R_u &= \left(\frac{a^2}{r}\right) e \sin E \\
R_n &= -\frac{2}{3} r + (U - U_o) R_u \\
R_{xN} &= (a^2/r) \left[a_{xN} - \cos (E + \omega)\right] \\
R_{yN} &= (a^2/r) \left[a_{yN} - \sin (E + \omega)\right] \\
R_c &= \left(2 \frac{R_n}{n}\right) (U - U_o)
\end{aligned}\right\} \qquad (3.5.29)$$

$$\left.\begin{aligned}
U_u &= (a^2/r)\sqrt{1 - e^2} \\
U_n &= (U - U_o) U_u \\
U_{xN} &= \frac{a^2}{r} \left\{(1 + \frac{r}{a}) \sin (E + \omega)\right. \\
&\qquad \left. + a_{xN} e \sin E \left[\frac{e^2 - (1 + \sqrt{1 - e^2}) e \cos E}{(1 + \sqrt{1 - e^2})^2 \sqrt{1 - e^2}}\right] - \frac{a_{yN}}{1 + \sqrt{1 - e^2}}\right\} \\
U_{yN} &= \frac{a^2}{r} \left\{-(1 + \frac{r}{a}) \cos (E + \omega)\right. \\
&\qquad \left. + a_{yN} e \sin E \left[\frac{e^2 - (1 + \sqrt{1 - e^2}) e \cos E}{(1 + \sqrt{1 - e^2})^2 \sqrt{1 - e^2}}\right] + \frac{a_{xN}}{1 + \sqrt{1 - e^2}}\right\} \\
U_c &= \frac{2 U_n}{n} (U - U_o).
\end{aligned}\right\} \qquad (3.5.30)$$

The quantities, r, a, a_{xN}, a_{yN}, e, n, U, $e \cos E$, $e \sin E$, $\sin (E + \omega)$, and $\cos (E + \omega)$ are obtained from MNREP subroutine.

When the slant range is observed, the residual and coefficients of Equation (3.5.28) are given by the expressions:

$$R = R_1 \tag{3.5.31}$$

$$C_{\underline{\Delta n}_{n}} = \underline{L}_c \cdot \underline{U}\, R_n + \underline{L}_c \cdot \underline{V}\, U_n \tag{3.5.32}$$

$$C\Delta a_{xN} = \underline{L}_c \cdot \underline{U}\, R_{xN} + \underline{L}_c \cdot \underline{V}\, U_{xN} \tag{3.5.33}$$

$$C\Delta a_{yN} = \underline{L}_c \cdot \underline{U}\, R_{yN} + \underline{L}_c \cdot \underline{V}\, U_{yN} \tag{3.5.34}$$

$$C_{\Delta U_o} = \underline{L}_c \cdot \underline{U}\, R_u + \underline{L}_c \cdot \underline{V}\, U_u \tag{3.5.35}$$

$$C_{\Delta\Omega} = \underline{L}_c \cdot \underline{V}\, r \cos i - \underline{L}_c \cdot \underline{W}\, r \sin i \cos u \tag{3.5.36}$$

$$C_{\Delta i} = \underline{L}_c \cdot \underline{W}\, r \sin u \tag{3.5.37}$$

$$C_{\Delta c''} = \underline{L}_c \cdot \underline{U}\, R_c + \underline{L}_c \cdot \underline{V}\, U_c \tag{3.5.38}$$

where R_1 is given by Equation (3.5.10), $\underline{L}_c$ is given by Equation (3.5.6), and the quantities $\underline{U}$, $\underline{V}$, $\underline{W}$, sin u, and cos u are again obtained by the MNREP subroutine.

When azimuth A is observed then $R = R_2$ as given by Equation (3.5.11), and the coefficients are obtained by replacing $\underline{L}_c$, in Equations (3.5.32) through (3.5.38), by $\underline{\hat{A}}$ Equation (3.5.13). Along with azimuth, elevation h is always observed, and in this case, $R = R_3$ as given by Equation (3.5.12), and the corresponding coefficients are obtained by replacing $\underline{L}_c$ in Equation (3.5.32) through (3.5.38) with $\underline{\tilde{D}}$ Equation(3.5.14).

If right ascension α is observed then $R = R_4$ as given by Equation (3.5.22), and the corresponding coefficients are obtained by replacing $\underline{L}_c$ with $\underline{A}$ Equation(3.5.24). Along with right ascension, declination δ is always observed and in this case, $R = R_5$, as given by Equation (3.5.23), and the corresponding coefficients are obtained by replacing $\underline{L}_c$ with $\underline{D}$ Equation(3.5.25).

If range rate, $\dot{\rho}$, is observed, the preliminary coefficients are computed from:

$$\left.\begin{aligned}
\dot{R}_u &= \sqrt{\mu}\,\frac{a^{\frac{5}{2}}}{r^3}\,(e\cos E - e^2)\\
\dot{R}_n &= \dot{r}/3 + (U - U_o)\,\dot{R}_u\\
\dot{R}_{xN} &= \left(\sqrt{\mu}\,\frac{a^{\frac{5}{2}}}{r^3}\right)\left[\sin(E+\omega) - a_{xN}\,e\sin E - a_{yN}\right]\\
\dot{R}_{yN} &= \left(\sqrt{\mu}\,\frac{a^{\frac{5}{2}}}{r^3}\right)\left[-\cos(E+\omega) - a_{yN}\,e\sin E + a_{xN}\right]\\
\dot{R}_c &= \frac{2\dot{R}_n}{n}\,(U - U_o)
\end{aligned}\right\}\quad(3.5.39)$$

$$\left.\begin{aligned}
\dot{U}_u &= -\sqrt{\mu}\,\frac{a^{\frac{5}{2}}}{r^3}\,\sqrt{1 - e^2}\;e\sin E\\
\dot{U}_n &= \frac{r\dot{v}}{3} + (U - U_o)\,\dot{U}_u\\
\dot{U}_{xn} &= \left(\sqrt{\mu}\,\frac{a^{\frac{5}{2}}}{r^3}\right)\sqrt{1 - e^2}\left[\cos(E+\omega) - a_{xN}\left(1 + \frac{r^2}{ap}\right)\right]\\
\dot{U}_{yn} &= \left(\sqrt{\mu}\,\frac{a^{\frac{5}{2}}}{r^3}\right)\sqrt{1 - e^2}\left[\sin(E+\omega) - a_{yN}\left(1 + \frac{r^2}{ap}\right)\right]\\
\dot{U}_c &= \frac{2\dot{U}_n}{n}\,(U - U_o),
\end{aligned}\right\}\quad(3.5.40)$$

then form the coefficients:

$$\begin{aligned} C_{\underset{n}{\underline{\Delta n}}} = {} & \underline{L}_c \cdot \underline{U} \left[\rho_c \, (\dot{R}_n - \dot{v} U_n) - \dot{\rho}_c \, R_n \right] + \dot{\underline{\rho}}_c \cdot \underline{U} \, R_n \\ & + \underline{L}_c \cdot \underline{V} \left[\rho_c (\dot{U}_n + \frac{\dot{r}}{r} U_n) - \dot{\rho}_c U_n \right] + \dot{\underline{\rho}}_c \cdot \underline{V} \, U_n \end{aligned} \qquad (3.5.41)$$

$$\begin{aligned} C\Delta a_{xN} = {} & \underline{L}_c \cdot \underline{U} \left[\rho_c \, (\dot{R}_{xN} - \dot{v} \, U_{xN}) - \dot{\rho}_c \, R_{xN} \right] + \dot{\underline{\rho}}_c \cdot \underline{U} \, R_{xN} \\ & + \underline{L}_c \cdot \underline{V} \left[\rho_c \, (\dot{U}_{xN} + \frac{\dot{r}}{r} U_{xN} - \dot{\rho}_c \, U_{xN} \right] + \dot{\underline{\rho}}_c \cdot \underline{V} \, U_{xN} \end{aligned} \qquad (3.5.42)$$

$$\begin{aligned} C\Delta a_{yN} = {} & \underline{L}_c \cdot \underline{U} \left[\rho_c \, (\dot{R}_{yN} - \dot{v} U_{yN}) - \rho_c \, R_{yN} \right] + \dot{\underline{\rho}}_c \cdot \underline{U} \, R_{yN} \\ & + \underline{L}_c \cdot \underline{V} \left[\rho_c \, (\dot{U}_{yN} + \frac{\dot{r}}{r} U_{yN}) - \dot{\rho}_c \, U_{yN} \right] + \dot{\underline{\rho}}_c \cdot \underline{V} \, U_{yN} \end{aligned} \qquad (3.5.43)$$

$$\begin{aligned} C\Delta U_o = {} & \underline{L}_c \cdot \underline{U} \left[\rho_c \, (\dot{R}_u - \dot{v} U_u) - \dot{\rho}_c \, R_u \right] + \dot{\underline{\rho}}_c \cdot \underline{U} \, R_u \\ & + \underline{L}_c \cdot \underline{V} \left[\rho_c \, (\dot{U}_u + \frac{\dot{r}}{r} U_u) - \dot{\rho}_c \, U_u \right] + \dot{\underline{\rho}}_c \cdot \underline{V} \, U_u \end{aligned} \qquad (3.5.44)$$

$$\begin{aligned} C\Delta\Omega = {} & - \rho_c \, \underline{L}_c \cdot \underline{U} \, r \, \dot{v} \cos i + \underline{L}_c \cdot \underline{V} \cos i \left[\rho_c \, \dot{r} - \dot{\rho}_c \, r \right] \\ & + \dot{\underline{\rho}}_c \cdot \underline{V} \, r \cos i + \underline{L}_c \cdot \underline{W} \sin i \left[\rho_c \, (r \, \dot{v} \sin u - \dot{r} \cos u) \right. \\ & \left. + \dot{\rho}_c \, r \cos u \right] - \dot{\underline{\rho}}_c \cdot \underline{W} \, r \sin i \cos u \end{aligned} \qquad (3.5.45)$$

$$C_{\Delta i} = \underline{L}_c \cdot \underline{W} \left[\rho_c \, (r \, \dot{v} \cos u + \dot{r} \sin u) - \dot{\rho}_c \, r \sin u\right] + \underline{\dot{\rho}}_c \cdot \underline{W} \, r \sin u \tag{3.5.46}$$

$$C_{\Delta c''} = \underline{L}_c \cdot \underline{U} \left[\rho_c \, (\dot{R}_c - \dot{v} \, U_c) - \dot{\rho}_c \, R_c\right] + \underline{\dot{\rho}}_c \cdot \underline{U} \, R_c$$

$$+ \underline{L}_c \cdot \underline{V} \left[\rho_c \, (\dot{U}_c + \frac{\dot{r}}{r} U_c - \dot{\rho}_c \, U_c\right] + \underline{\dot{\rho}}_c \cdot \underline{V} \, U_c \, , \tag{3.5.47}$$

where R_6, given by Equation (3.5.27), is the corresponding residual for the above coefficients. The correction equation is formed as an Equation (3.5.28).

Considering all of the correction Equations (3.5.28) let:

$$\sum_{i=1}^{N} C_{ij} \; \Delta_i = R_{ij}$$

represent all such equations where: C_{ij}'s are the coefficients, R_{ij}'s are the accepted observation residuals, Δ_i's are the corrections to the orbital elements, and N is the number of elements to be corrected and j is the number of accepted observation residuals. The resulting matrix equation is solved to give the corrections, Δ_i's, in a least square sense, to the orbital elements at time, t_o. These corrections are applied as follows (primes denote corrected elements):

$$n_o' = n_o \, (1 + \frac{\Delta n_o}{n_o}) \tag{3.5.48}$$

$$(c'')' = c'' + \Delta c'' \tag{3.5.49}$$

$$c_o' = - \frac{(c'')'}{n_o^2} \, (\frac{\pi^2}{360}) \tag{3.5.50}$$

$$U_o' = U_o + \Delta U_o \tag{3.5.51}$$

$$a_{xNo}' = a_{xNo} + \Delta a_{xN} \tag{3.5.52}$$

$$a'_{yNo} = a_{yNo} + \Delta a_{yN} \tag{3.5.53}$$

$$\Omega'_o = \Omega_o + \Delta\Omega \tag{3.5.54}$$

$$i'_o = i_o + \Delta i \tag{3.5.55}$$

$$\begin{cases} L'_o = U'_o + \Omega'_o, \text{ if } W'_z = \cos i' \geq 0 \\ L'_o = U'_o - \Omega'_o, \text{ if } W'_z = \cos i' < 0 \end{cases} \tag{3.5.56}$$

Following the above calculation of the corrected elements, a representation of the observations is performed on the basis of the corrected elements, and another set of residuals are formed by using the same input observations. The RMS values of the last two sets of consecutive residuals are compared to insure convergence of the computational process. The process is complete when the residual RMS converges to the minimum value considered as acceptable.

3.5.3 SGPDC PROGRAM SPECIFICATION SUMMARY

a. System Input

COBLOC

EBLOC

SBLOC

DCCCP1

DCCCP2

DCCCP3

DCCCP4

DCCCP5

DCCCP6

DCCCP7

DCCCP8

DCCCP9

DIVCNT

DCFIRST

CCARD

LPOBLOC

KCOUNT

b. Associated Input Cards

E CARD

S CARD

SGPDC CONTROL

c. System Output Data

VEBLOC
DEBLOC
DIVCNT
KCOUNT
CCARD
DCFIRST

d. System Common Subroutine

ARCTAN
ELOUTB
COBSGET
DELPCK
EUNPCK
ELMPCK
GLOP
IELMPCK
IDELPCK
INITCOB
MNREP
NXJOB
EXEND
ELOUT A
PAGECON
PANT
SENGET
CHGNXN
SXSRCH
CARDER
SETJC
LSQ
LSQR
LSQS

e. Philco Subroutines

FSIN
FCOS
FMAIN
FASIN
FSQRT
FMAMU
FLOG2X
F2X

3.5.4 SGPDC SCHEMATIC COMPUTATIONAL SEQUENCE

SGPDC operations are expanded in Figures 3-11 through 3-14 to show the detailed processes. All the major computations given in the formulation are indicated where applicable in the diagrams, and any additional computations are shown as needed.

The following glossary serves as a centralized source for the definition of storage assignments used in the diagrams:

ABSMAX	Absolute maximum of range and angle residuals override compiled as 1000 km.
ABSMAX 1	Second absolute maximum of range and angle residuals override, compiled as 75 km.
ABSMAX 2	Absolute maximum of range rate residuals override, compiled as 0.5 km/sec.
ABSMAX 3	Second absolute maximum of range and angle RMS residuals test, compiled as 50 km.
ACCCNT	Accumulated number of good residuals.
CONTEST	Maximum value acceptable for convergence compiled as 0.05 percent.
DCCCP 1	Elements to be corrected override as specified on SGPDC control card.
COUNTL	Number of elements to be corrected as found in the current KNTRL.
DIVFL	Divergence flag, indicates if new element set has lead to a larger residual RMS than the previous element set.
KNTRL	Indicates which elements are to be corrected as specified by KTABLE or DCCCP 1.
KTABLE	Compiled specification of element sets to be corrected in given sequence.
OLDSUM	Next to last converging RMS residual value.
ORRMS	Last converging RMS residual value.
RCNT	Number of good range and angle type residuals.
RCNT 2	Number of good range rate residuals.
REJCNT	Number of rejected residuals.

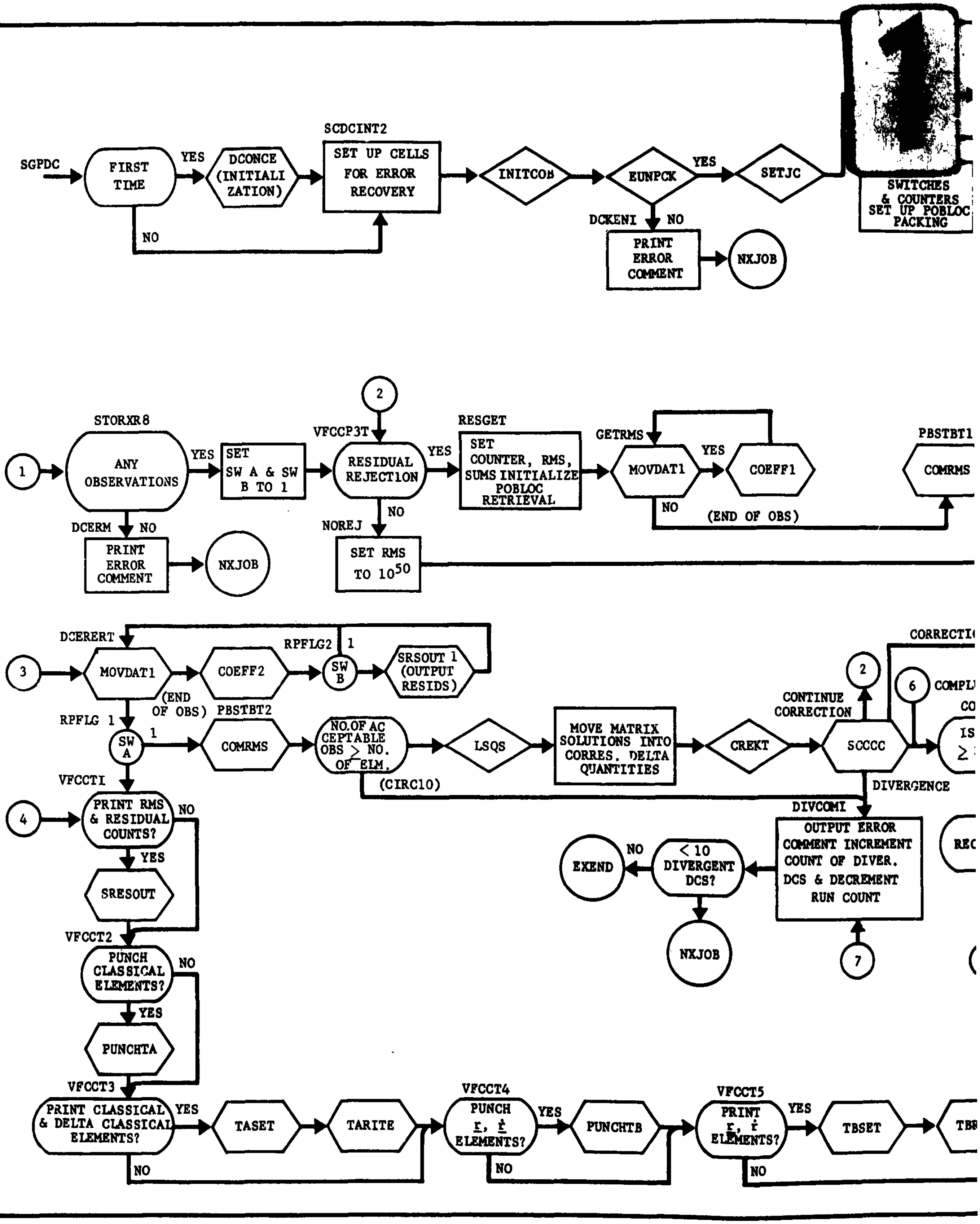

SGPDC
FIRST TIME
YES
NO
DCONCE (INITIALI ZATION)
SCDCINT2
SET UP CELLS FOR ERROR RECOVERY
INITCOB
EUNPCK
YES
SETJC
SWITCHES & COUNTERS SET UP POBLOC PACKING
DCKENI
NO
PRINT ERROR COMMENT
NXJOB
1
STORXR8
ANY OBSERVATIONS
YES
SET SW A & SW B TO 1
2
VFCCP3T
RESIDUAL REJECTION
YES
RESGET
SET COUNTER, RMS, SUMS INITIALIZE POBLOC RETRIEVAL
GETRMS
MOVDAT1
YES
COEFF1
PBSTBT1
COMRMS
NO
(END OF OBS)
DCERM
NO
PRINT ERROR COMMENT
NXJOB
NO
NOREJ
SET RMS TO 10^{50}
DCERERT
3
MOVDAT1
COEFF2
RPFLG2
1
SW B
SRSOUT 1 (OUTPUT RESIDS)
(END OF OBS)
PBSTBT2
RPFLG 1
SW A
1
COMRMS
NO. OF AC CEPTABLE OBS $\geq$ NO. OF ELM.
LSQS
MOVE MATRIX SOLUTIONS INTO CORRES. DELTA QUANTITIES
CREKT
CORRECTI
2
CONTINUE CORRECTION
6
COMPL
CO
SCCCC
IS $\geq$
(CIRC10)
DIVERGENCE
DIVCOMI
VFCCT1
4
PRINT RMS & RESIDUAL COUNTS?
NO
YES
SRESOUT
OUTPUT ERROR COMMENT INCREMENT COUNT OF DIVER. DCS & DECREMENT RUN COUNT
REC
EXEND
NO
< 10 DIVERGENT DCS?
NXJOB
7
VFCCT2
PUNCH CLASSICAL ELEMENTS?
NO
YES
PUNCHTA
VFCCT3
PRINT CLASSICAL & DELTA CLASSICAL ELEMENTS?
YES
NO
TASET
TARITE
VFCCT4
PUNCH $\underline{r}$, $\dot{\underline{r}}$ ELEMENTS?
YES
NO
PUNCHTB
VFCCT5
PRINT $\underline{r}$, $\dot{\underline{r}}$ ELEMENTS?
YES
NO
TBSET
TBR

FIGURE 3-11. S

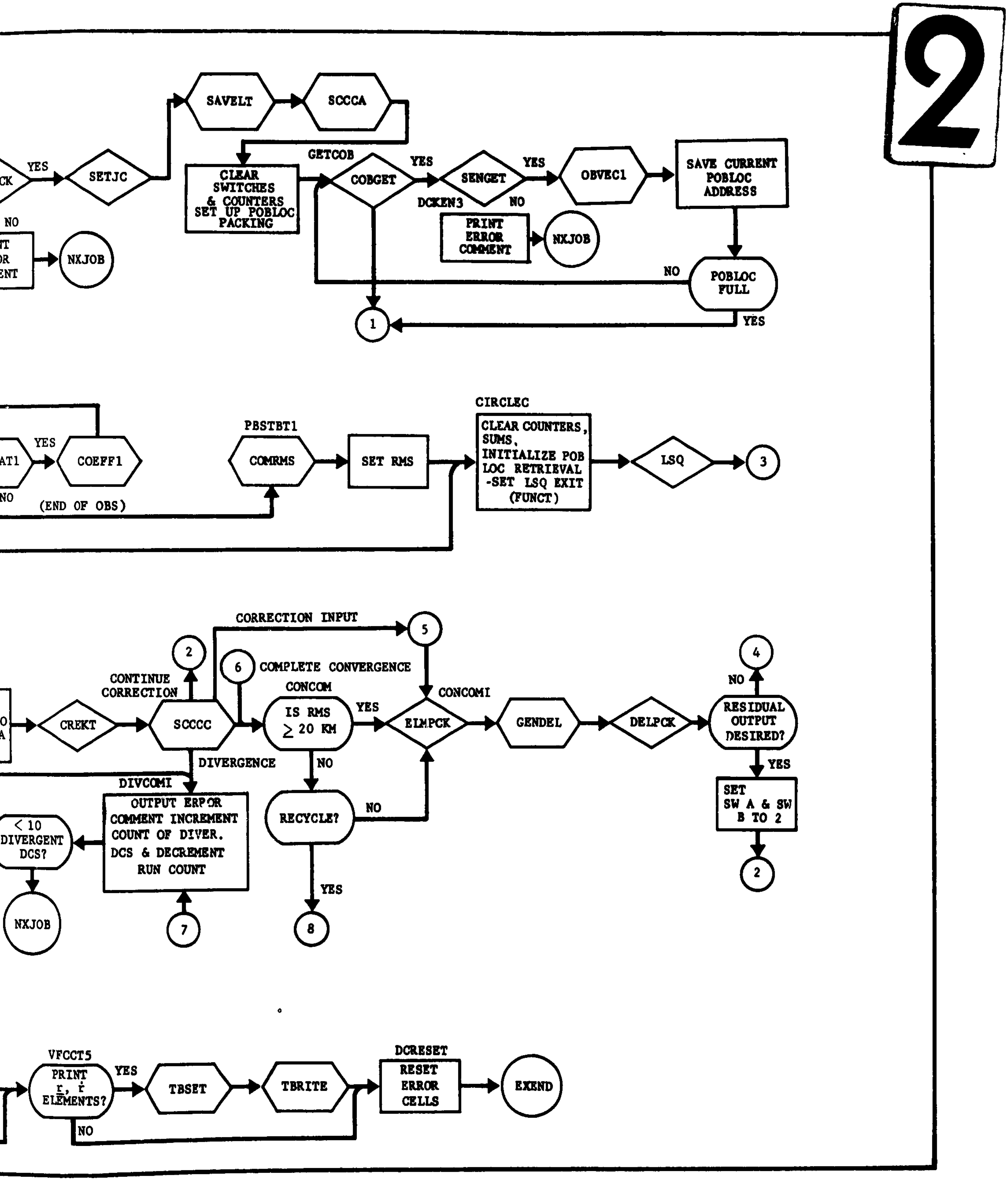
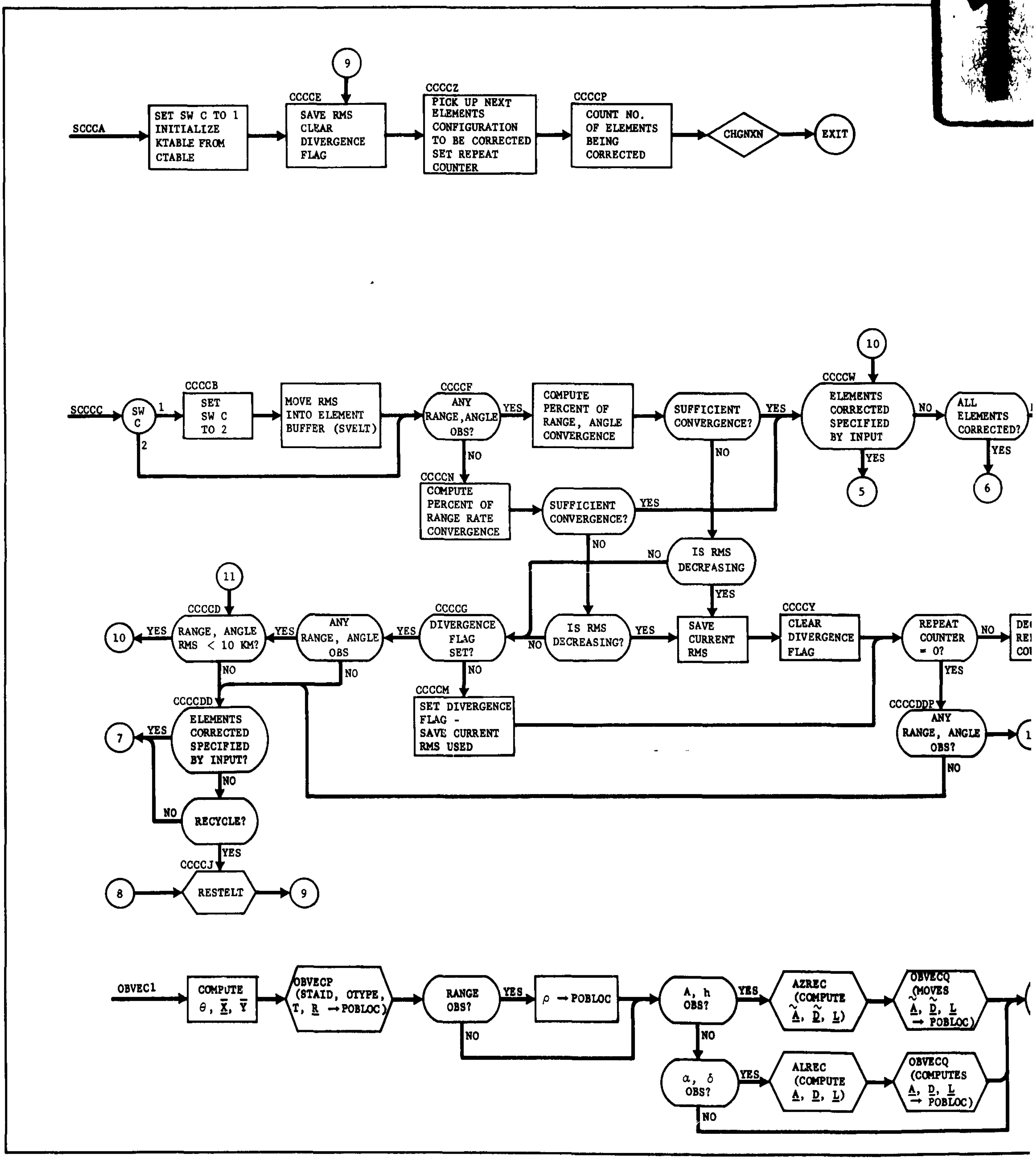


FIGURE 3-11. SGPDC


9
SCCCA
SET SW C TO 1 INITIALIZE KTABLE FROM CTABLE
CCCCE
SAVE RMS CLEAR DIVERGENCE FLAG
CCCCZ
PICK UP NEXT ELEMENTS CONFIGURATION TO BE CORRECTED SET REPEAT COUNTER
CCCCP
COUNT NO. OF ELEMENTS BEING CORRECTED
CHGNXN
EXIT
10
SCCCC
SW C
1
2
CCCCB
SET SW C TO 2
MOVE RMS INTO ELEMENT BUFFER (SVELT)
CCCCF
ANY RANGE, ANGLE OBS?
YES
NO
COMPUTE PERCENT OF RANGE, ANGLE CONVERGENCE
SUFFICIENT CONVERGENCE?
YES
NO
CCCCW
ELEMENTS CORRECTED SPECIFIED BY INPUT
NO
YES
5
ALL ELEMENTS CORRECTED?
YES
6
CCCCN
COMPUTE PERCENT OF RANGE RATE CONVERGENCE
SUFFICIENT CONVERGENCE?
YES
NO
IS RMS DECREASING
NO
YES
11
CCCCD
10
YES
RANGE, ANGLE RMS < 10 KM?
YES
ANY RANGE, ANGLE OBS
YES
CCCCG
DIVERGENCE FLAG SET?
NO
IS RMS DECREASING?
YES
SAVE CURRENT RMS
CCCCY
CLEAR DIVERGENCE FLAG
REPEAT COUNTER = 0?
NO
DE(RE COI
NO
NO
NO
CCCCDD
7
YES
ELEMENTS CORRECTED SPECIFIED BY INPUT?
CCCCM
SET DIVERGENCE FLAG - SAVE CURRENT RMS USED
YES
CCCCDDP
ANY RANGE, ANGLE OBS?
1
NO
NO
NO
RECYCLE?
YES
CCCCJ
8
RESTELT
9
OBVEC1
COMPUTE θ, X̄, Ȳ
OBVECP (STAID, OTYPE, T, R → POBLOC)
RANGE OBS?
YES
ρ → POBLOC
NO
A, h OBS?
YES
AZREC (COMPUTE Ã, D̃, L)
OBVECQ (MOVES Ã, D̃, L → POBLOC)
NO
α, δ OBS?
YES
ALREC (COMPUTE A, D, L)
OBVECQ (COMPUTES A, D, L → POBLOC)
NO

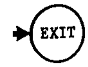
FIGURE

	41	42	43	44	45	46	47
	C''	i	Ω	U_o	a_{xn}	a_{yn}	n
C TABLE		1	1	1	1	1	1
+1				1			1
+2			1	1			1
+3		1	1	1			1
+4		1	1	1	1	1	1
C TABLE	1	1	1	1	1	1	1
+1				1			1
+2	1			1			1
+3	1		1	1			1
+4	1	1	1	1			1
+5	1	1	1	1	1	1	1

(BITS IN C TABLE)

ASSEMBLED CORRECTION TABLE (WITHOUT DRAG)

ASSEMBLED CORRECTION TABLE (WITH DRAG)



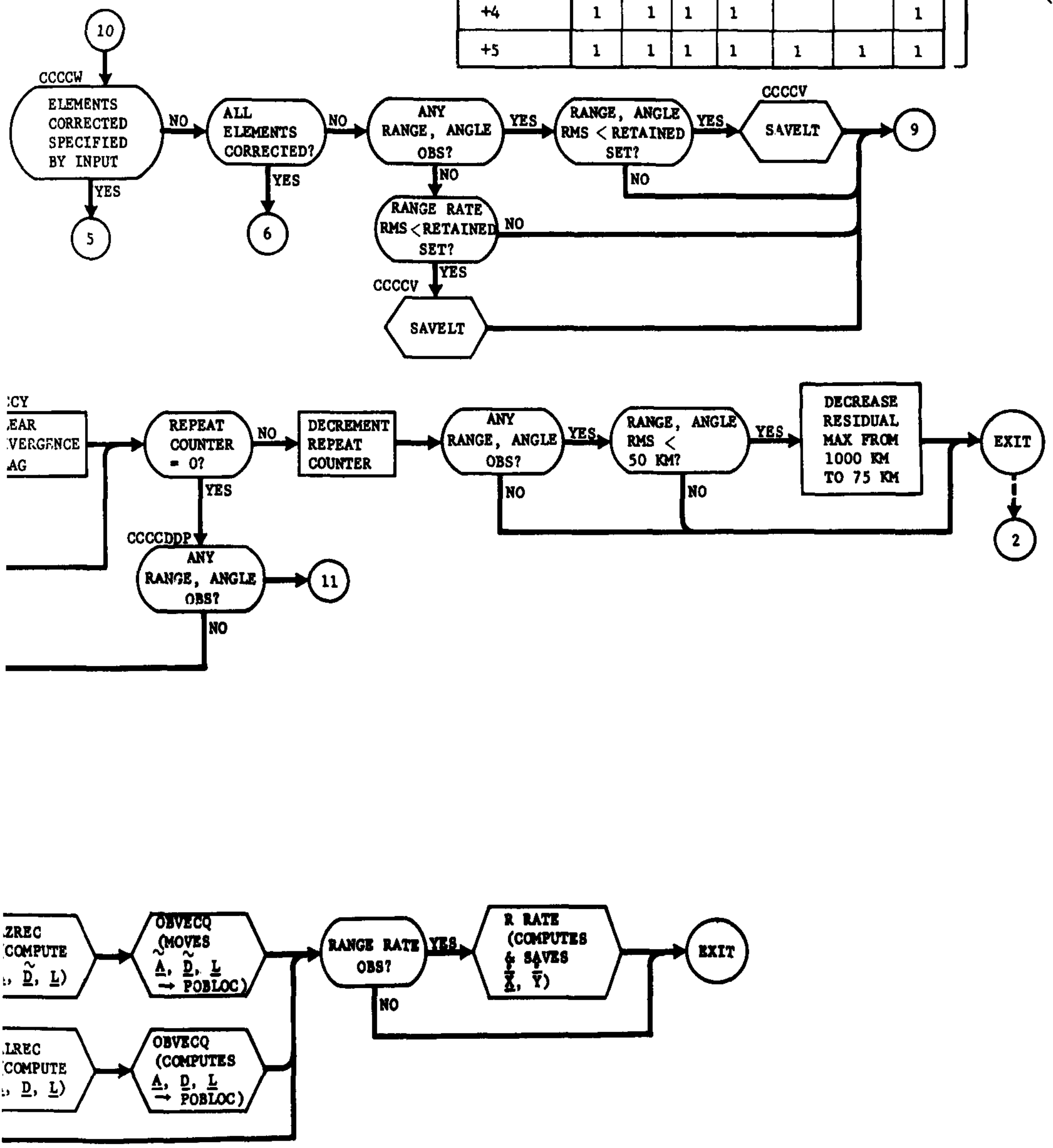


FIGURE 3-12. SGPDC (Continued)

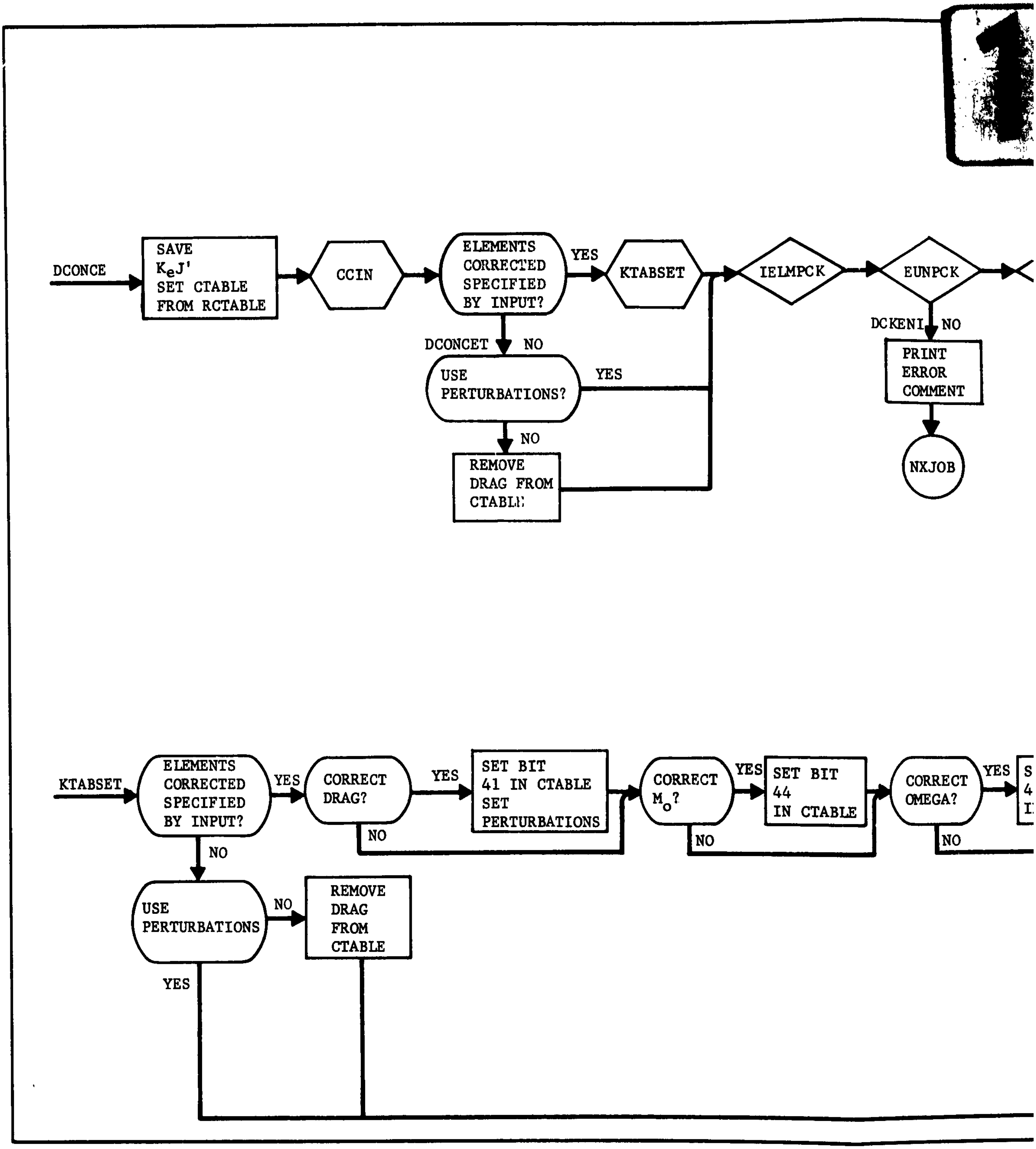

1
DCONCE
SAVE
K_eJ'
SET CTABLE
FROM RCTABLE
CCIN
ELEMENTS
CORRECTED
SPECIFIED
BY INPUT?
YES
KTABSET
IELMPCK
EUNPCK
DCKENI
NO
PRINT
ERROR
COMMENT
NXJOB
DCONCET
NO
USE
PERTURBATIONS?
YES
NO
REMOVE
DRAG FROM
CTABLE
KTABSET
ELEMENTS
CORRECTED
SPECIFIED
BY INPUT?
YES
CORRECT
DRAG?
YES
SET BIT
41 IN CTABLE
SET
PERTURBATIONS
NO
CORRECT
M_o?
YES
SET BIT
44
IN CTABLE
NO
CORRECT
OMEGA?
YES
S
4
I
NO
NO
USE
PERTURBATIONS
NO
REMOVE
DRAG
FROM
CTABLE
YES

FIGURE 3-1

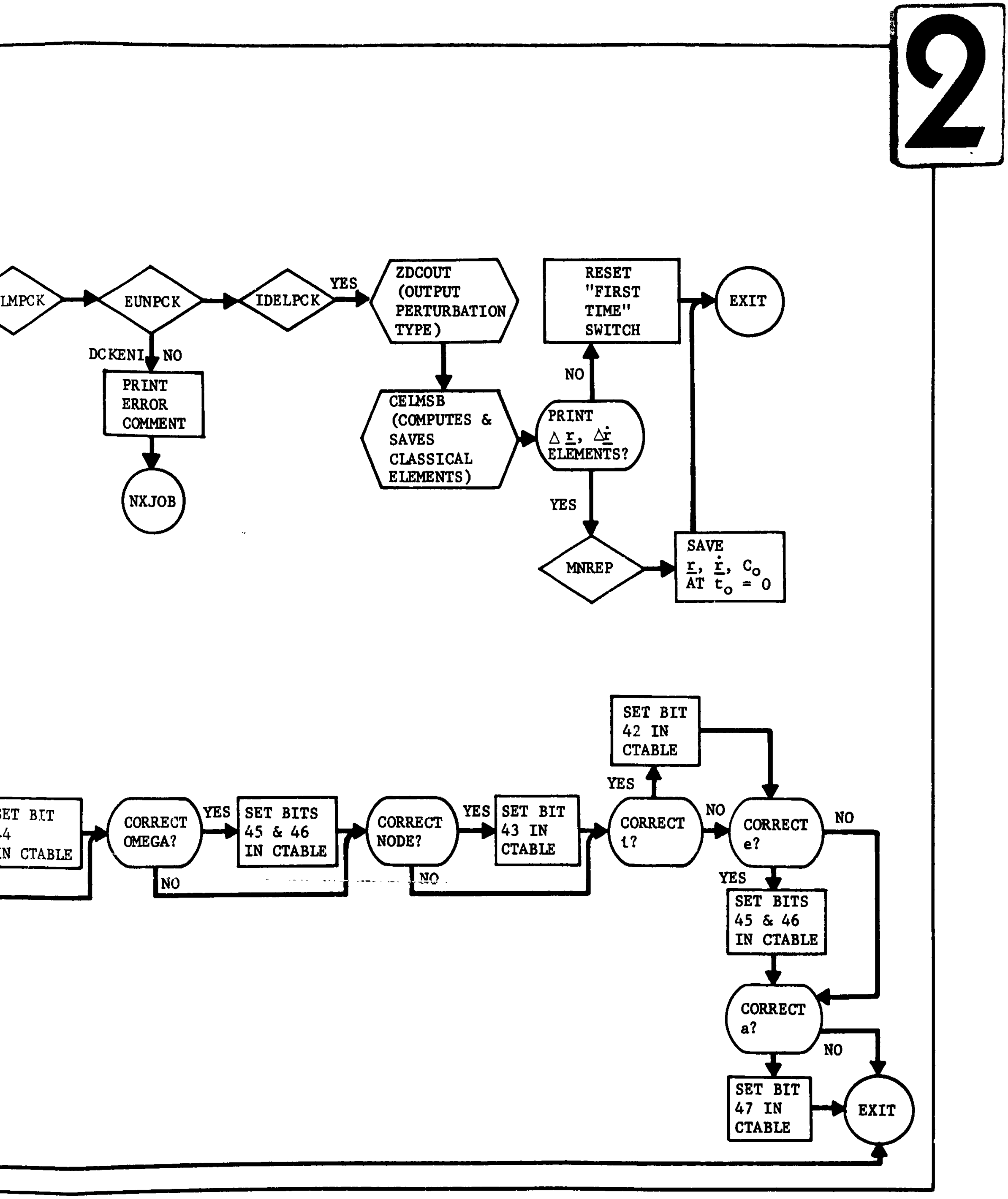


FIGURE 3-13. SGPDC (Continued)

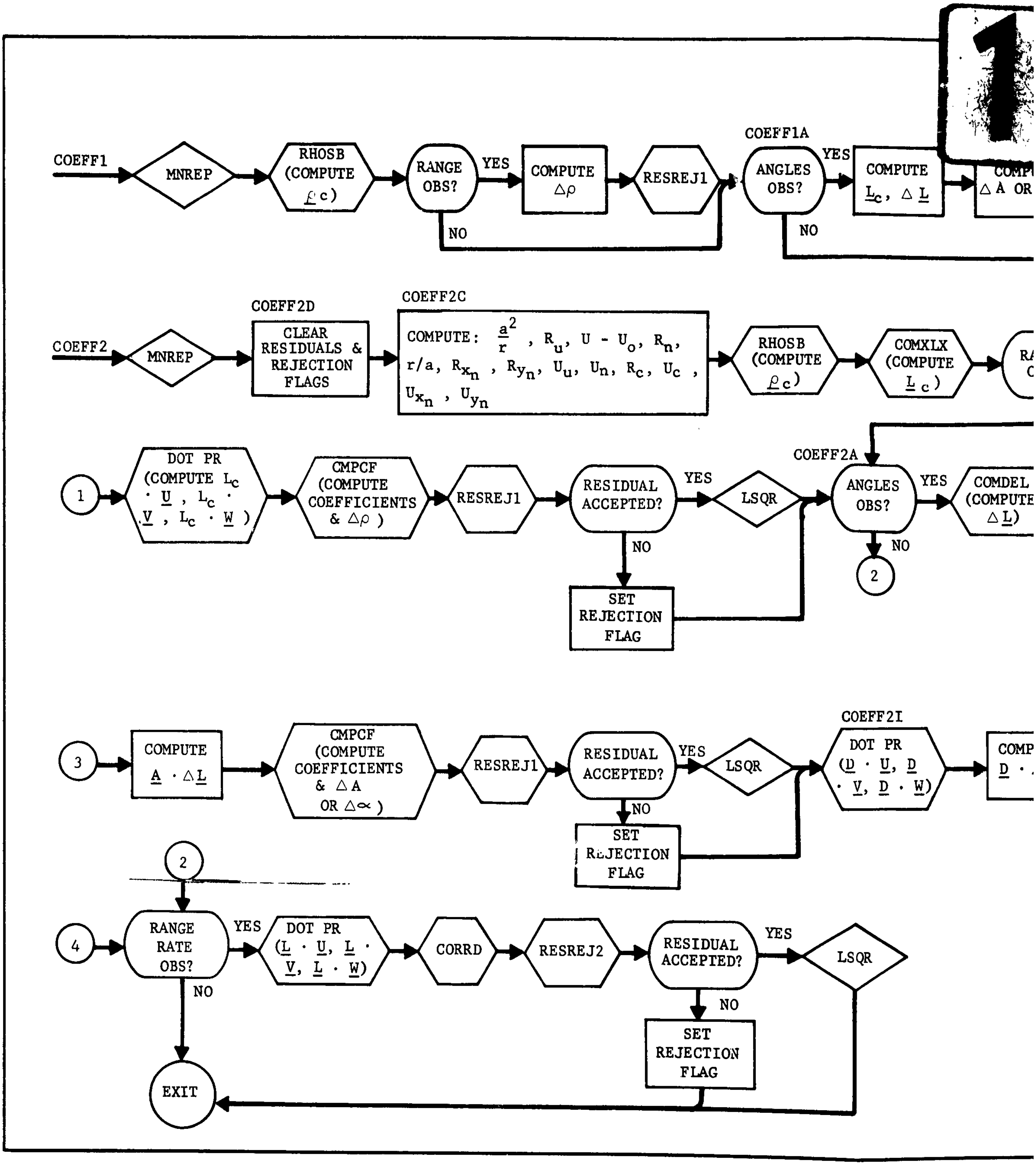

COEFF1
MNREP
RHOSB (COMPUTE ρc)
RANGE OBS?
YES
NO
COMPUTE Δρ
RESREJ1
COEFF1A
ANGLES OBS?
YES
NO
COMPUTE Lc, ΔL
COMP ΔA OR
COEFF2
MNREP
COEFF2D
CLEAR RESIDUALS & REJECTION FLAGS
COEFF2C
COMPUTE: a²/r, Ru, U - Uo, Rn, r/a, Rxn, Ryn, Uu, Un, Rc, Uc, Uxn, Uyn
RHOSB (COMPUTE ρc)
COMXLX (COMPUTE Lc)
RA C
1
DOT PR (COMPUTE Lc · U, Lc · V, Lc · W)
CMPCF (COMPUTE COEFFICIENTS & Δρ)
RESREJ1
RESIDUAL ACCEPTED?
YES
NO
LSQR
SET REJECTION FLAG
COEFF2A
ANGLES OBS?
YES
NO
2
COMDEL (COMPUTE ΔL)
3
COMPUTE A · ΔL
CMPCF (COMPUTE COEFFICIENTS & ΔA OR Δα)
RESREJ1
RESIDUAL ACCEPTED?
YES
NO
LSQR
SET REJECTION FLAG
COEFF2I
DOT PR (D · U, D · V, D · W)
COMP D ·
2
4
RANGE RATE OBS?
YES
NO
DOT PR (L · U, L · V, L · W)
CORRD
RESREJ2
RESIDUAL ACCEPTED?
YES
NO
LSQR
SET REJECTION FLAG
EXIT

FIGURE 3-14

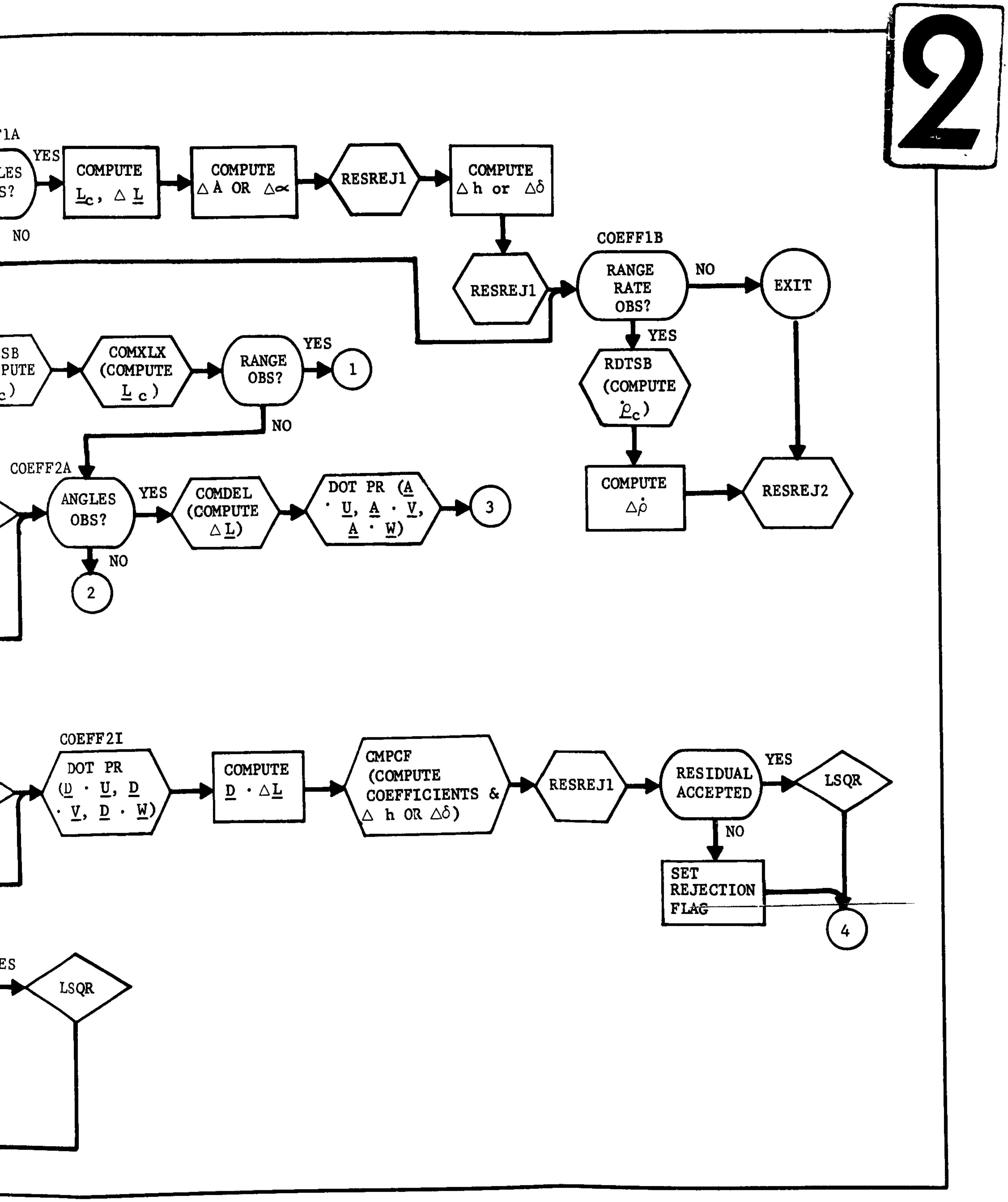


FIGURE 3-14. SGPDC (Continued)

RPT Number of iterations per element correction specification.

SUM Sum of squares of good range and angle residuals.

MAX Current maximum value of acceptable range and angle residual RMS.

MAX 2 Current maximum value of acceptable range rate residual RMS.

XISTSG RMS multiplication factor.

XJGRCF Perturbation control.

ALREC Computes $\underline{A}$, $\underline{D}$ and $\underline{L}$ from right ascension and declination observations.

AZREC Computes $\underline{\tilde{A}}$, $\underline{\tilde{D}}$ and $\underline{L}$ from azimuth and elevation observations.

CCIN Tests the overrides input on the SGPDC control card and initializes the DC accordingly. CCIN uses SETJC to initialize perturbations.

CELMSB Converts $\underline{N}$, $\underline{M}$ elements to classical and stores them in CEBLOC (element type A and B only). The following formulas are used:

$$P_o = h_{x_o}^2 + h_{y_o}^2 + h_{z_o}^2$$

$$\underline{W}_o = \frac{\underline{h}_o}{\sqrt{P_o}}$$

$$i_o = \tan^{-1} \frac{\sqrt{1 - W_z^2}}{W_z}$$

$$\Omega_o = \tan^{-1} \frac{W_{x_o}}{W_{y_o}}$$

$$e_o^2 = a_{xN_o}^2 + a_{yN_o}^2$$

$$a_o = \frac{P_o}{1 - e_o^2}$$

$$\omega_o = \tan^{-1} \frac{a_{yN_o}}{a_{xN_o}}$$

$$M_o = L_o \mp \Omega_o - \omega_o$$

- if $W_z \geq 0$, direct motion

+ if $W_z < 0$, retrograde motion

$$c_o \equiv c_o$$

CMPCF Computes coefficients C_i (of Equation 3.5.28) for range and angle observations.

COMDEL Computes $\Delta\underline{L}$, $\Delta\underline{L} = \underline{L} - \underline{L}_c$ with $\underline{L}$ from Equation 3.5.14 or 3.5.26.

COMRMS Computes RMS of acceptable observations.

COMXLX Computes $\underline{L}_c$, Equation 3.5.6.

CORRD Computes coefficients C_i (Equation 3.5.28) for range rate observations.

CREKT Computes the corrected elements (Equation 3.5.48).

DELTAU Computes geocentric angle between observed and computed satellite position vector, ΔU.

$$\Delta u = \tan^{-1} \left[\frac{(\underline{U} \times \underline{r}_{obs}) \cdot \underline{W}}{\underline{U} \cdot \underline{r}_{obs}} \right]$$

DOTPR Computes vector dot products of $\underline{U}$, $\underline{V}$ and $\underline{W}$ with any of $\underline{L}_c$, $\underline{A}$, $\dot{\underline{A}}$, $\underline{D}$ or $\dot{\underline{D}}$.

GENDEL Generates delta classical elements according to the following formulas:

$$\omega'_o = \tan^{-1} \frac{a'_{yN_o}}{a'_{xN_o}}$$

$$M'_o = L'_o \mp \Omega'_o - \omega'_o \quad \begin{cases} -\Omega'_o & \text{if } W_z \geq 0 \\ +\Omega'_o & \text{if } W_z < 0 \end{cases}$$

$$\Delta a_o = a'_o - a_o$$

$$\Delta e_o = e'_o - e_o$$

$$\Delta i_o = i'_o - i_o$$

$$\Delta\Omega_o = \Omega'_o - \Omega'_o$$

$$\Delta\omega_o = \omega'_o - \omega_o$$

$$\Delta M_o = M'_o - M_o$$

$$\Delta c_o = c'_o - c_o$$

Note: Prime indicates variant element set.
All Δ angles are mod 2π.
If the corresponding element has no correction specified, the delta element is set = 0.

MOVDATI Retrieves one processed observation from POBLOC. MOVDATI has a special return for end of observations.

OBVEC Computes θ, X, Y.

OBVECP Moves station number, observation type, time since epoch, and X, Y, Z into POBLOC.

OBVECQ Moves $\underline{A}$ or $\underline{\tilde{A}}$, $\underline{D}$ or $\underline{\tilde{D}}$ and $\underline{L}$ into POBLOC.

PUNCHTA Punches Type A variant element cards 1 and 2. (See card formats).

PUNCHTB Uses common subroutine MNREP to compute $\underline{r}$, $\dot{\underline{r}}$ elements at epoch and punches Type B variant element cards 1 and 2. (See card formats).

RDTSB Computes $\dot{\underline{\rho}}_c$, $\dot{\rho}_c$, Equations 3.5.8 and 3.5.9.

RESREJ1 Rejects range and angle observation if residual is greater than MAX and accumulates a count of acceptable and unacceptable observations; also, computes the sum of squares of acceptable residuals.

RESREJ2 Same as RESREJ1 except rejects range rate observation greater than MAX2.

RESTELT Restores an element set from the SVELT buffer. (See SAVELT for contents).

RHOSB Computes ρ_c, Equation 3.5.3.

RRATE Computes $\dot{R}$, Equation 3.5.7.

SAVELT Saves a corrected element set in SVELT buffer. This includes the following: a_{xN_o}, a_{yN_o}, h_{x_o}, h_{y_o}, h_{z_o}, L_o, c_o, rms for range and angles, rms for range rate, $\frac{\Delta n}{n}$, Δa_{xN}, Δa_{yN}, ΔU_o, $\Delta\Omega$, Δi, $\Delta c''$, number of acceptable residuals, number of residuals rejected, number of elements corrected.

SRESOUT Uses program subroutine COMRMS to compute the RMS's and outputs them in addition to the count of the residuals accepted and rejected.

SRSOUTI Performs the following:

(1) Computes vector magnitude

(2) Computes r_{obs}

(3) Uses DELTAU (program subroutine) to compute Δu.

(4) Computes: $U = U_c + \Delta u$, $\Delta t = \dfrac{r^2 \Delta u}{k_e \sqrt{p}}$,

$$\beta = \sin^{-1}\left(\frac{\underline{r}_{obs} \cdot \underline{W}}{r_{obs}}\right)$$

(5) Outputs: Station ID, time residuals, rejection flags, vector magnitude, Δt, U, β.

TARITE Uses common subroutine ELOUTA to print classical elements in EBUF.

TASET Saves classical elements: a_o, e_o, i_o, Ω_o, ω_o, M_o, C_o, t_o in EBUF for output.

TBRITE Uses ELOUTB to print $\underline{r}$, $\underline{\dot{r}}$ elements in EBUF and prints $\Delta\underline{r}$, $\Delta\underline{\dot{r}}$ elements in EBUFD.

TBSET Uses common subroutine MNREP to compute $\underline{r}$, $\underline{\dot{r}}$ at epoch and saves the following in EBUF: x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$, C_o, t_o. Also computes $\Delta\underline{r}$, $\Delta\underline{\dot{r}}$ elements and saves them in EBUFD.

ZDCOUT Outputs comments indicating perturbation type used in SGPDC.

3.6 ELEMENT DISTRIBUTION - DCDIST

Statistical processing of the delta elements generated by the Differential Correction Program Unit is the function of the DCDIST Program Unit. The delta elements generated by SGPDC are the differences between the variant element sets and the reference element set. These differences represent, within the sensor error model chosen, errors in the determination of orbital elements.

3.6.1 DCDIST INPUT-OUTPUT DATA

The input data, DEBLOC, consist of K sets of the quantities,

$$\Delta a, \Delta e, \Delta i, \Delta\Omega, \Delta\omega, \Delta M_o, \text{ and possibly } \Delta C_o.$$

The DCDIST control parameters, input via the SPGDC control card, consist of:

DCNS The number of samples of an element error used to compute the mean and standard deviation error.

DCI The number of cells in each element frequency distribution.

DCS1 The number of standard deviations below the mean at the first cell of the element frequency distributions.

DCS2 The number of standard deviations above the mean at the last cell of the element frequency distributions.

DCCCP1 Control parameter from SGPDC indicating which elements have simulated errors.

DCCCP3 Control parameter from SGPDC indicating perturbation analysis option.

DCCCP10 Classical element distribution suppression.

The output feom DCDIST routine contains a printed listing of the following statistical data:

Identification of error quantity

Mean

Standard deviation

Minimum error, MIN

Maximum error, MAX

Summation of DCNS errors

Summation of the square of DCNS errors

DCNS

Total number of errors, N

DCS1

DCS2

Frequency distribution

Lower limit of the frequency distribution

Cell width of the frequency distribution

3.6.2 DCDIST FORMULATION

The presentation of the computations in the DCDIST Unit is given under Common Subroutine, STOB.

3.6.3 DCDIST PROGRAM SPECIFICATION SUMMARY

a. System Input

DEBLOC

DCS1

DCS2

DCI

DCNS

DECOUNT

DCCCP1

DCCCP3

DCCCP10

b. Associated Input Cards

SPGDC CONTROL

MOCARO CONTROL

c. System Output

For each of the element errors, Δa, Δe, Δi, $\Delta\Omega$, $\Delta\omega$, ΔM_o, ΔC_o, the following output is produced:

MU

SIGMA

MIN

MAX

Sum of Errors

Sum of Squared Errors

Freq Dist Lower Limit

Freq Dist Cell Width

Freq Dist

d. System Common Subroutines

STOB

EXEND

SPOUT1

PANT

e. Philco Subroutines

(None)

3.6.4 DCDIST SCHEMATIC COMPUTATIONAL SEQUENCE

Figure 3-15 displays the information processing for DCDIST using the STOB subroutine. Details of the STOB routine are available in the subroutine section.

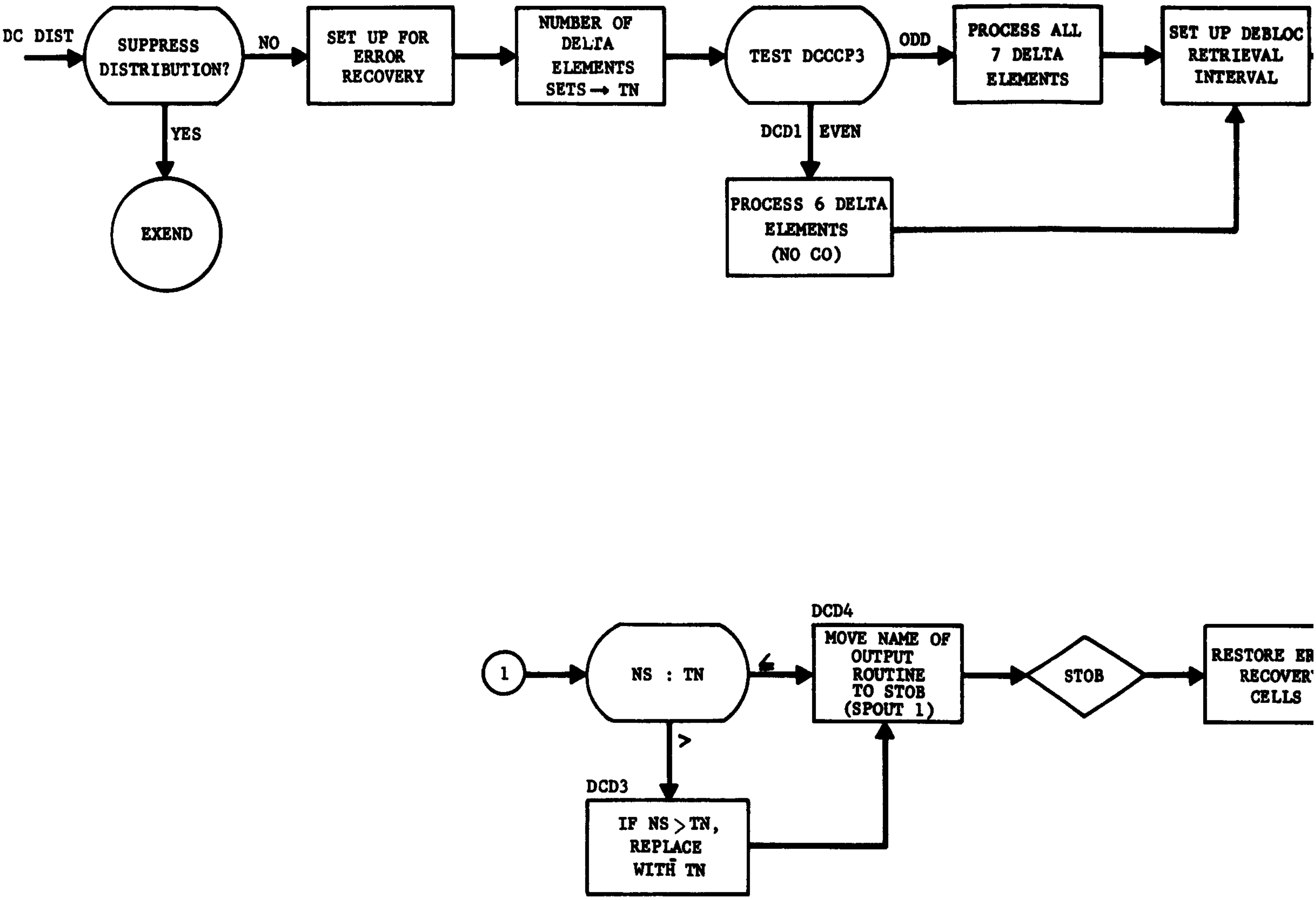

DC DIST
SUPPRESS DISTRIBUTION?
NO
YES
EXEND
SET UP FOR ERROR RECOVERY
NUMBER OF DELTA ELEMENTS SETS → TN
TEST DCCCP3
ODD
DCD1
EVEN
PROCESS ALL 7 DELTA ELEMENTS
SET UP DEBLOC RETRIEVAL INTERVAL
PROCESS 6 DELTA ELEMENTS (NO CO)
1
NS : TN
>
DCD3
IF NS > TN, REPLACE WITH TN
DCD4
MOVE NAME OF OUTPUT ROUTINE TO STOB (SPOUT 1)
STOB
RESTORE ER RECOVER CELLS

FIGURE 3-15. I

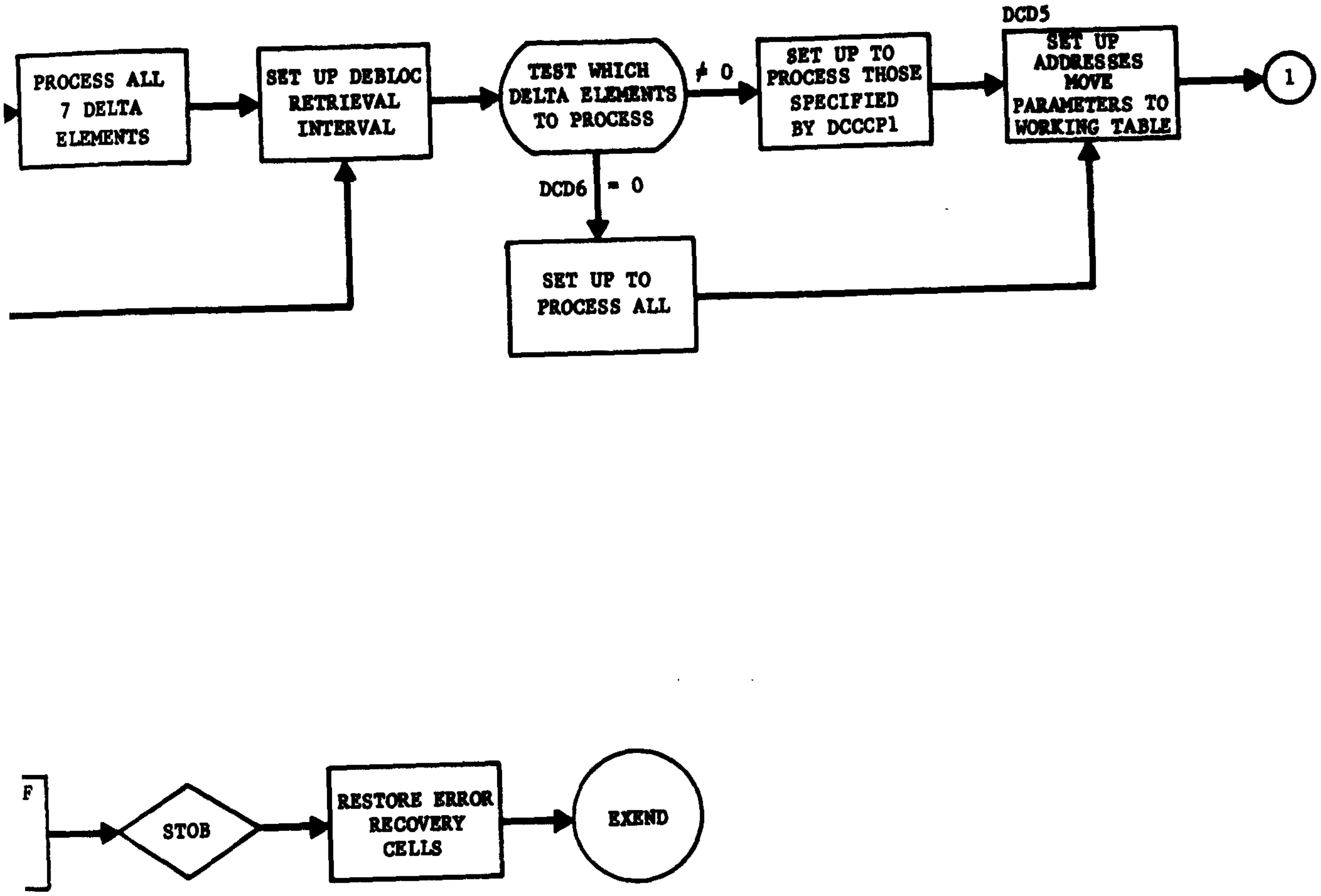

PROCESS ALL 7 DELTA ELEMENTS
SET UP DEBLOC RETRIEVAL INTERVAL
TEST WHICH DELTA ELEMENTS TO PROCESS
≠ 0
SET UP TO PROCESS THOSE SPECIFIED BY DCCCP1
DCD5
SET UP ADDRESSES MOVE PARAMETERS TO WORKING TABLE
1
DCD6
= 0
SET UP TO PROCESS ALL
F
STOB
RESTORE ERROR RECOVERY CELLS
EXEND

FIGURE 3-15. DCDIST

3.7 ERROR PREDICTION - ERP

The purpose of the Error Prediction Program Unit is to compute the errors in position and velocity at specified points due to variations in the orbit elements established in the MOCARO-SGPDC loop. The unit is functionally connected to the Prediction Distribution Program Unit and the two units are referred to in the executive logic as ERPRED. This functional connection is necessary because of the quadrant, or segment, prediction analysis specification for the system. In response to this system function, the two units were designed to provide compatible interfaces and thus act as two large subroutines under the control of an executive driver. However, for logical presentation of the system components, these two units are discussed separately.

Several modes of operation are available for the Error Prediction Program Unit. These likewise apply to the combined ERPRED function. The modes of operation are:

(1) Specification of points for evaluation of errors, designated by Casetype, an integer from 0 to 3.

(a) Casetype = 0: The prediction points are equally spaced in mean anomaly, M, and are referenced to the first perigee passage after epoch. P_i = initial M, $\Delta P = \Delta M$, and P_f = final N. Note: If the eccentricity is too small, then these data are automatically assumed to be U data, mean latitude and a printout is made indicating this.

(b) Casetype = 2: There are N equally spaced points in each of the four quadrants of mean anomaly, M, following the time of the last observation. $P_i = N$. (ΔP and P_f not used).

(c) Casetype = 3: The prediction points are equally spaced in time referenced to epoch. P_i = initial time, $P = \Delta t$, P_f = final time.

(2) Specification of form of output

(a) If OTPFLAG = 0, then the position and velocity errors are referenced to the x, y, z, coordinate system.

(b) If OTPFLAG ≠ 0, then the position and velocity errors are referenced to the U, V, W coordinate system.

(3) Specification of Method of Computation

(a) If APMFLAG = 0, then the errors are computed by representing the variant orbits using MNREP and subtracting the reference orbit position and velocity. In this mode, the perturbation control on MNREP is that assigned to the system in initialization.

(b) If APMFLAG = 1, then the errors are computed using MNREP with perturbations, overriding the system control.

(c) If APMFLAG = 2, then the errors are computed using MNREP without perturbations, overriding the system control.

(d) If APMFLAG = 3, then the errors are computed by multiplying the errors in the element set by a matrix of partial derivatives (Linear Taylor's expansion).

3.7.1 ERP INPUT-OUTPUT DATA

The input system and control data include:

(1) Operating Mode Indicators

CASETYP

APMFLAG

OTPFLAG

(2) Prediction Points

P_i, ΔP, P_f

Number of variant element sets = k

(3) Reference Element Set

L_o, a_{xN_o}, a_{yN_o}, h_{x_o}, h_{y_o}, h_{z_o}, t_o(EPOCH), C_o

(4) Variant Element Sets

DEBLOC (Δa_j, Δe_j, Δi_j, $\Delta\Omega_j$, $\Delta\omega_j$, ΔM_{oj}, ΔC_{oj})

VEBLOC (L_{oj}, $a_{xN_{oj}}$, $a_{yN_{oj}}$, $h_{x_{oj}}$) $h_{y_{oj}}$, $h_{z_{oj}}$)

The reference element set, delta element sets, and variant element sets are available from stored data blocks, EBLOC, DEBLOC, and VEBLOC, respectively. The remaining input data enter the routine via the error prediction card.

The output data are contained in DPVBLOC and consist of either delta position and velocity in equatorial coordinates

$\Delta x, \Delta y, \Delta z, \Delta\dot{x}, \Delta\dot{y}, \Delta\dot{z}, |\Delta \underline{r}|, |\Delta\dot{\underline{r}}|$

or orbital coordinates

$D_u, D_v, D_w, \dot{D}_u, \dot{D}_v, \dot{D}_w, |\Delta\underline{r}|, |\Delta\dot{\underline{r}}|$

3.7.2 ERP FORMULATION

(1) Compute e_o^2:

$$e_o^2 = a_{xN_o}^2 + a_{yN_o}^2 \tag{3.7.1}$$

(2) Compute ω_o and Ω_o:

$$\omega_o = \tan^{-1} \frac{a_{yN_o}}{a_{xN_o}}* \tag{3.7.2}$$

*In expressions using the arctan, the quadrant is determined by inspection of the numerator and denominator.

$$\Omega_o = \tan^{-1} \frac{h_{x_o}}{-h_{y_o}} \qquad (3.7.3)$$

(3) Compute Ω_o:

$$\Omega_o = \tan^{-1} \frac{h_{x_o}}{-h_{y_o}} \qquad (3.7.4)$$

(4) Compute t_i, Δt, t_f from P_i: (N is stored in P_i)

t_{OBF} = time of last observation

$$t_i = t_{OBF} - t_o \qquad (3.7.5)$$

$$p_o = h_{x_o}^{\ 2} + h_{y_o}^{\ 2} + h_{z_o}^{\ 2} \qquad (3.7.6)$$

$$a_o = p_o / (1 - e_o^{\ 2}) \qquad (3.7.7)$$

$$\Delta t = \frac{\pi a_o^{\ 3/2}}{2\ P_i k_e \sqrt{\mu}} \qquad (3.7.8)$$

$$t_f = \frac{2\ \pi a_o^{\ 3/2}}{k_e \sqrt{\mu}} + t_{OBF} - t_o \qquad (3.7.9)$$

(5) Initial Quadratic Coefficients for U (a, b, and c):

$$n_o = \frac{k_e \sqrt{\mu}}{a_o^{\ 3/2}} \qquad (3.7.10)$$

(a) If XJGRCF = 0 (no perturbations), then

$$a = 0$$

$$b = n_o$$

$$c = L_o - P_i - \frac{h_{z_o}}{|h_{z_o}|} \Omega_o$$

(b) If XJGRCF ≠ 0 (perturbations), then

$$c'' = - \frac{360 n_o^2 c_o}{\pi^2}$$

$$J' = \sqrt{\mu} \frac{J a_e^2}{p_o^{7/2}}$$

$$a = c'' \left\{ n_o + k_e J' \frac{|h_{z_o}|}{\sqrt{p_o}} 4 e_o (1 - e_o) \right\}$$

$$b = n_o + k_e J' \left[-1 + \frac{7}{4} e_o^2 + \frac{h_{z_o}^2}{p_o} (4 - \frac{27}{4} e_o^2) \right]$$

$$c = L_o - P_i - \frac{h_{z_o}}{|h_{z_o}|} \Omega_o$$

(6) Initial Quadratic Coefficients for M (a, b, and c):

(a) If XJGRCF = 0 (no perturbations), then

$$a = 0$$

$$b = n_o$$

$$c = L_o - P_i - \omega_o - \frac{h_{z_o}}{|h_{z_o}|} \Omega_o$$

(b) If XJGRCF ≠0 (perturbations), then

$$c'' = - \frac{360 \, n_o^2 \, c_o}{\pi^2}$$

$$J' = \frac{\sqrt{\mu} \, J \, a_e^2}{p_o^{7/2}}$$

$$a = c'' \left[n_o + 2k_e J' e_o \, (1-e_o) \left\{ 1+2 \, \frac{|h_{z_o}|}{\sqrt{p_o}} - 5 \, \frac{h^2_{z_o}}{p_o} \right\} \right]$$

$$b = n_o - k_e J' \left(\frac{1}{2} - e_o^2 \right) \left(1-3 \, \frac{h^2_{z_o}}{p_o} \right)$$

$$c = L_o - P_i - \omega_o - \frac{h_{z_o}}{|h_{z_o}|} \Omega_o$$

(7) Solve quadratic equation for t_n:

(a) If $\frac{|4ac|}{b^2} \geq \delta_2$, then

$$t_n = - \frac{b}{2a} \left\{ 1 - \sqrt{1 - \frac{4ac}{b^2}} \right\} \qquad (3.7.11)$$

(b) If $\left| \frac{4ac}{b^2} \right| \leq \delta_2$, then

$$t_n = - \frac{c}{b} \left(1 + \frac{ac}{b^2}\right) \qquad (3.7.12)$$

(c) If $(1 - \frac{4ac}{b^2}) \leq 0$ then

$$t_n = - \frac{b}{2a} \tag{3.7.13}$$

Print: NEGATIVE RADICAND EQUALS - .rrr ± ee
Continue processing

(8) Compute Partial Derivative Matrix $[\,p_{ij}\,]$, where

$p_{11} = R_m$

$p_{12} = R_{xN}$

$p_{13} = R_{yN}$

$p_{14} = \frac{R_a}{a}$

$p_{15} = p_{16} = 0$

$p_{21} = V_m$

$p_{22} = V_{xN}$

$p_{23} = V_{yN}$

$p_{24} = \frac{V_a}{a}$

$p_{25} = 0$

$p_{26} = r \cos i$

$p_{31} = p_{32} = p_{33} = p_{34} = 0$

$p_{35} = r \sin u$

$$p_{36} = -r \sin i \cos u$$

$$p_{41} = \dot{R}_m - \dot{v}\, V_m$$

$$p_{42} = \dot{R}_{xN} - \dot{v}\, V_{xN}$$

$$p_{43} = \dot{R}_{yN} - \dot{v}\, V_{yN}$$

$$p_{44} = (\dot{R}_a - \dot{v}\, V_a) / a$$

$$p_{45} = 0$$

$$p_{46} = - r\, \dot{v} \cos i$$

$$p_{51} = \dot{V}_m + \frac{\dot{r}}{r} V_m$$

$$p_{52} = \dot{V}_{xN} + \frac{\dot{r}}{r}\; V_{xN}$$

$$p_{53} = \dot{V}_{yN} + \frac{\dot{r}}{r}\; V_{yN}$$

$$p_{54} = (\dot{V}_a + \frac{\dot{r}}{r} V_a) / a$$

$$p_{55} = 0$$

$$p_{56} = \dot{r} \cos i$$

$$p_{61} = p_{62} = p_{63} = p_{64} = 0$$

$$p_{65} = r\, \dot{v} \cos u + \dot{r} \sin u$$

$$p_{66} = (r\, \dot{v} \sin u - \dot{r} \cos u) \sin i$$

The auxiliary quantities in the p_{ij} elements are obtained from:

$$R_m = \frac{a^2}{r} e \sin E \qquad (3.7.14)$$

$$R_a = r - \frac{3}{2} (U - U_o) R_m \qquad (3.7.15)$$

$$R_{xN} = \frac{a^2}{r} \left[a_{xN} - \cos (E + \omega) \right] \qquad (3.7.16)$$

$$R_{yN} = \frac{a^2}{r} \left[a_{yN} - \sin (E + \omega) \right] \qquad (3.7.17)$$

$$V_m = \frac{a^2}{r} - \sqrt{1-e^2} \qquad (3.7.18)$$

$$V_a = - \frac{3}{2} (U - U_o) V_m \qquad (3.7.19)$$

$$V_{xN} = \frac{a^2}{r} \left[(1 + \frac{r}{a}) \sin (E + \omega) \right. \qquad (3.7.20)$$

$$+ a_{xN} e \sin E \frac{e^2 - (1 + \sqrt{1-e^2}) e \cos E}{\sqrt{1-e^2} (1 + \sqrt{1-e^2})^2}$$

$$\left. - \frac{a_{yN}}{1 + \sqrt{1-e^2}} \right]$$

$$V_{yN} = \frac{a^2}{r} \left[- (1 + \frac{r}{a}) \cos (E + \omega) \right. \qquad (3.7.21)$$

$$+ a_{yN} e \sin E \frac{e^2 - (1 + \sqrt{1-e^2}) e \cos E}{\sqrt{1 - e^2} (1 + \sqrt{1-e^2})^2}$$

$$\left. + \frac{a_{xN}}{1 + \sqrt{1-e^2}} \right]$$

$$\dot{R}_m = \sqrt{\mu} a^{5/2} (e \cos E - e^2) / r^3 \qquad (3.7.22)$$

$$\dot{R}_a = -\frac{1}{2}\dot{r} - \frac{3}{2}(U - U_o)\dot{R}_m \qquad (3.7.23)$$

$$\dot{R}_{xN} = \sqrt{\mu}\, a^{5/2} \left[\sin (E + \omega) - a_{xN}\, e \sin E - a_{yN}\right] / r^3 \qquad (3.7.24)$$

$$\dot{R}_{yN} = \sqrt{\mu}\, a^{5/2} \left[-\cos (E + \omega) - a_{yN}\, e \sin E + a_{xN}\right] / r^3 \qquad (3.7.25)$$

$$\dot{V}_m = \sqrt{\mu}\, a^{5/2} \sqrt{1-e^2}\, e \sin E / r^3 \qquad (3.7.26)$$

$$\dot{V}_a = -\frac{1}{2}\, r\, \dot{v} - \frac{3}{2}(U - U_o)\dot{V}_m \qquad (3.7.27)$$

$$\dot{V}_{xN} = \sqrt{\mu}\, a^{5/2} \sqrt{1-e^2} \left[\cos (E+\omega) - a_{xN}\left(1+ \frac{r^2}{ap}\right)\right] / r^3 \qquad (3.7.28)$$

$$\dot{V}_{yN} = \sqrt{\mu}\, a^{5/2} \sqrt{1-e^2} \left[\sin (E+\omega) - a_{yN}\left(1+ \frac{r^2}{ap}\right)\right] / r^3 \qquad (3.7.29)$$

The parameters which are used in the above expressions are available from the MNREP subroutine.

(9) Compute U, V, W errors using Partial Matrix:

$$\left|\begin{array}{c} D_u \\ D_v \\ D_w \\ D_u \\ D_v \\ D_w \end{array}\right|_{j,n} = \left[\begin{array}{c} \\ \\ P \\ \\ \\ \end{array}\right]_n \times \left|\begin{array}{c} \Delta U_o \\ \Delta a_{xN} \\ \Delta a_{yN} \\ \Delta a \\ \Delta i \\ \Delta\Omega \end{array}\right|_j \qquad (3.7.30)$$

(10) Convert U, V, W errors to x, y, z errors:

$$\left.\begin{aligned}
\Delta x &= D_u U_x + D_v V_x + D_w W_x \\
\Delta y &= D_u U_y + D_v V_y + D_w W_y \\
\Delta z &= D_u U_z + D_v V_z + D_w W_z \\
\Delta \dot{x} &= \dot{D}_u U_x + \dot{D}_v V_x + \dot{D}_w W_x \\
\Delta \dot{y} &= \dot{D}_u U_y + \dot{D}_v V_y + \dot{D}_w W_y \\
\Delta \dot{z} &= \dot{D}_u U_z + \dot{D}_v V_z + \dot{D}_w W_z
\end{aligned}\right\} \tag{3.7.31}$$

(11) Convert x, y, z errors to U, V, W errors:

$$\left.\begin{aligned}
D_u &= \Delta x\, U_x + \Delta y\, U_y + \Delta z\, U_z \\
D_v &= \Delta x\, V_x + \Delta y\, V_y + \Delta z\, V_z \\
D_w &= \Delta x\, W_x + \Delta y\, W_y + \Delta z\, W_z \\
\dot{D}_u &= \Delta \dot{x}\, U_x + \Delta \dot{y}\, U_y + \Delta \dot{z}\, U_z \\
\dot{D}_v &= \Delta \dot{x}\, V_x + \Delta \dot{y}\, V_y + \Delta \dot{z}\, V_z \\
\dot{D}_w &= \Delta \dot{x}\, W_x + \Delta \dot{y}\, W_y + \Delta \dot{z}\, W_z
\end{aligned}\right\} \tag{3.7.32}$$

(12) Compute $|\Delta \underline{r}|$, $|\Delta \dot{\underline{r}}|$:

$$|\Delta \underline{r}| = \left[\Delta x^2 + \Delta y^2 + \Delta z^2\right]^{1/2} \tag{3.7.33}$$

$$|\Delta \dot{\underline{r}}| = \left[\Delta \dot{x}^2 + \Delta \dot{y}^2 + \Delta \dot{z}^2\right]^{1/2} \tag{3.7.34}$$

or

$$|\Delta r| = \left[D_u^{\ 2} + D_v^{\ 2} + D_w^{\ 2} \right]^{1/2} \tag{3.7.35}$$

$$|\Delta \dot{r}| = \left[\dot{D}_u^{\ 2} + \dot{D}_v^{\ 2} + \dot{D}_w^{\ 2} \right]^{1/2} \tag{3.7.36}$$

Those equations are used for which the data are available.

3.7.3 ERP PROGRAM SPECIFICATION SUMMARY

(a) System Input

EBLOC

DEBLOC

VEBLOC

CASETYP

OTPFLG

APMFLG

DPVFLG

CAPPI

DELCAPP

CAPPF

CCARD

ERPOPT

LDPBLOC

PERT

LDEBLOC

LVEBLOC

VECOUNT

LOCOBS

(b) Associated Input Cards

ELEMENT

ERPRED CONTROL

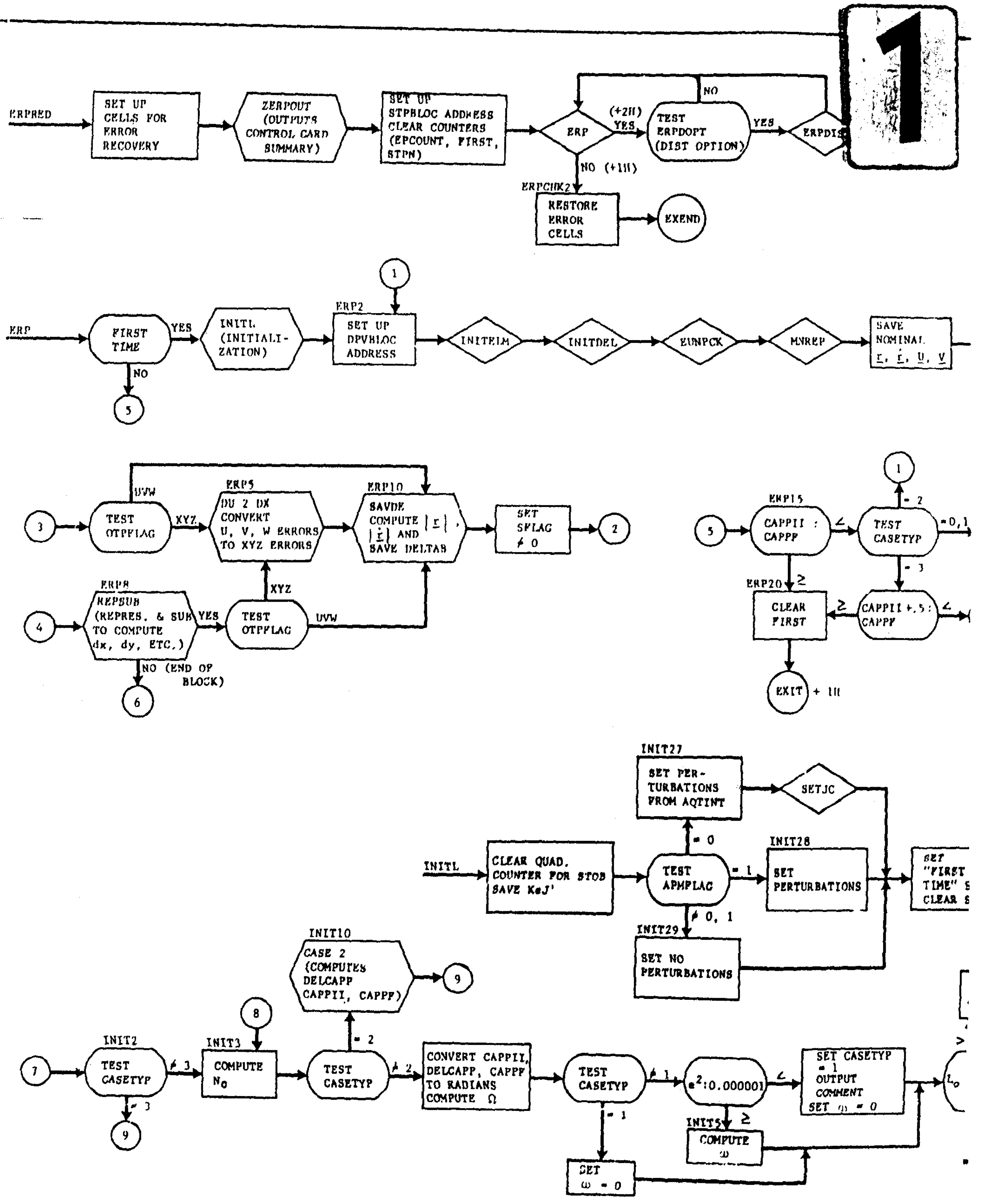
(c) System Output

DPVBLOC

EPCOUNT

CCARD

STPSTB



1
ERPRED
SET UP CELLS FOR ERROR RECOVERY
ZERPOUT (OUTPUTS CONTROL CARD SUMMARY)
SET UP STPBLOC ADDRESS CLEAR COUNTERS (EPCOUNT, FIRST, STPN)
ERP
(+2H) YES
NO (+1H)
TEST ERPDOPT (DIST OPTION)
YES
NO
ERPDIS
ERPCHK2
RESTORE ERROR CELLS
EXEND
ERP
FIRST TIME
YES
NO
5
INITL (INITIALIZATION)
ERP2
1
SET UP DPVBLOC ADDRESS
INITEIM
INITDEL
EUNPCK
MNREP
SAVE NOMINAL r, ṙ, U, V
3
TEST OTPFLAG
UVW
XYZ
ERP5
DU 2 DX CONVERT U, V, W ERRORS TO XYZ ERRORS
ERP10
SAVDE COMPUTE |r|, |ṙ| AND SAVE DELTAS
SET SFLAG ≠ 0
2
4
ERP8
REPSUB (REPRES. & SUB TO COMPUTE dx, dy, ETC.)
YES
NO (END OF BLOCK)
6
TEST OTPFLAG
XYZ
UVW
5
ERP15
CAPPII : CAPPF
<
TEST CASETYP
= 2
1
=0,1
= 3
ERP20
≥
CLEAR FIRST
≥
CAPPII +.5 : CAPPF
<
EXIT + 1H
INIT27
SET PERTURBATIONS FROM AQTINT
SETJC
INITL
CLEAR QUAD. COUNTER FOR STOB SAVE KeJ'
TEST APHFLAG
= 0
= 1
INIT28
SET PERTURBATIONS
≠ 0, 1
INIT29
SET NO PERTURBATIONS
SET "FIRST TIME" & CLEAR
INIT10
CASE 2 (COMPUTES DELCAPP CAPPII, CAPPF)
9
7
INIT2
TEST CASETYP
≠ 3
= 3
9
INIT3
8
COMPUTE N₀
TEST CASETYP
= 2
≠ 2
CONVERT CAPPII, DELCAPP, CAPPF TO RADIANS COMPUTE Ω
TEST CASETYP
≠ 1
= 1
SET ω = 0
e²:0.000001
<
≥
INIT5
COMPUTE ω
SET CASETYP = 1 OUTPUT COMMENT SET ω = 0
L₀

FIGURE 3-

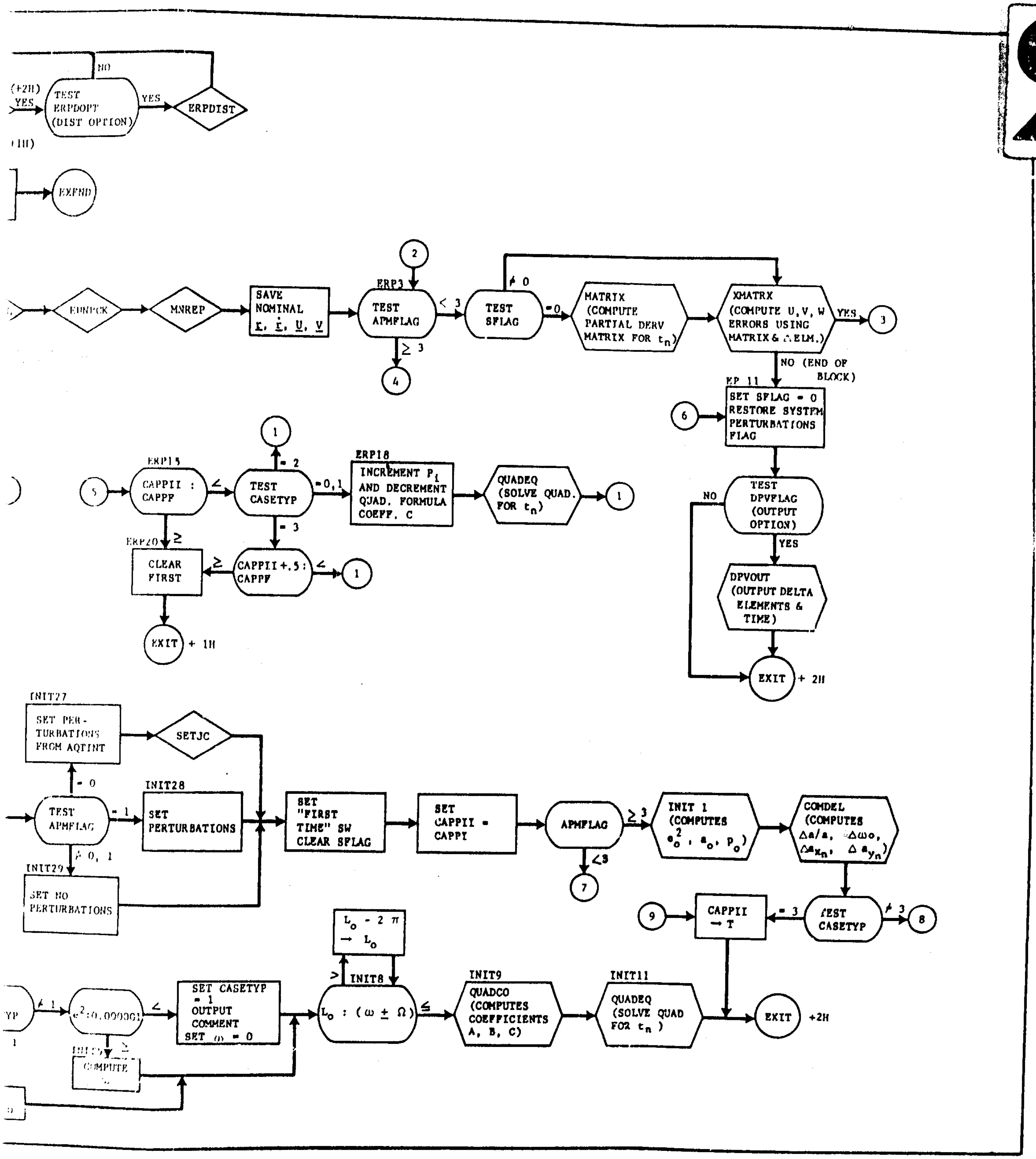

NO
(+2H)
YES
TEST ERPDOPT (DIST OPTION)
YES
ERPDIST
+1H)
EXFND
EDNPCK
MNREP
SAVE NOMINAL r, ṙ, U, V
2
ERP3
TEST APMFLAG
< 3
≥ 3
4
TEST SFLAG
≠ 0
= 0
MATRIX (COMPUTE PARTIAL DERV MATRIX FOR t_n)
XMATRX (COMPUTE U,V, W ERRORS USING MATRIX & ΔELM.)
YES
3
NO (END OF BLOCK)
EP 11
SET SFLAG = 0 RESTORE SYSTEM PERTURBATIONS FLAG
6
TEST DPVFLAG (OUTPUT OPTION)
NO
YES
DPVOUT (OUTPUT DELTA ELEMENTS & TIME)
EXIT +2H
1
ERP15
5
CAPPII : CAPPF
<
TEST CASETYP
= 2
= 0,1
= 3
ERP18
INCREMENT P_i AND DECREMENT QUAD. FORMULA COEFF. C
QUADEQ (SOLVE QUAD. FOR t_n)
1
ERP20
≥
CLEAR FIRST
≥
CAPPII+.5 : CAPPF
<
1
EXIT + 1H
INIT27
SET PERTURBATIONS FROM AQTINT
SETJC
= 0
TEST APMFLAG
= 1
INIT28
SET PERTURBATIONS
≠ 0, 1
INIT29
SET NO PERTURBATIONS
SET "FIRST TIME" SW CLEAR SFLAG
SET CAPPII = CAPPI
APMFLAG
≥ 3
<3
7
INIT 1 (COMPUTES e_o^2, a_o, p_o)
COMDEL (COMPUTES Δa/a, Δωo, $Δa_{x_n}$, $Δa_{y_n}$)
TEST CASETYP
≠ 3
8
= 3
CAPPII → T
9
$L_o - 2\pi \rightarrow L_o$
>
INIT8
YP
≠ 1
e^2:0.000001
<
SET CASETYP = 1 OUTPUT COMMENT SET ω = 0
≥
COMPUTE
L_o : (ω ± Ω)
≤
INIT9
QUADCO (COMPUTES COEFFICIENTS A, B, C)
INIT11
QUADEQ (SOLVE QUAD FOR t_n)
EXIT +2H

FIGURE 3-16. ERPRED

(d) System Common Subroutines

ARCTAN
DELGET
ELMGET
EUNPCK
GLOP
IELMGET
INITDEL
MNREP
PAGECON
PANT
EXEND
NXJOB
INITELM
SETJC
SXSRCH
CARDER

(e) Philco Subroutines

FSQRT

3.7.4 ERP SCHEMATIC COMPUTATIONAL SEQUENCE

The Error Prediction diagram showing the program logic is presented in Figure 3-16. A glossary of the main internal subroutines used in the diagram is given below for reference:

MATRIX Computes the coefficients of the matrix (Step 8).

XMATRX Solves Equation 3.7.30.

DU2DX Converts D_u, D_v, D_w errors to Δx, Δy, Δz using Equations 3.7.31.

SAVDE Computes $|\Delta \underline{r}|$ and $|\Delta \dot{\underline{r}}|$ using Equations 3.7.33, 3.7.34 or Equations 3.7.35, 3.7.36, storing in DPVBLOC

QUADEQ Computes t_n using Equations 3.7.11, 3.7.12, 3.7.13.

CASE2 Computes t_i, Δt, t_f (Equations 3.7.5, 3.7.8, 3.7.9).

QUADCO Computes a, b, c for Equations 3.7.11, 3.7.12, 3.7.13.

INIT1 Computes e_o^2, a_o, p_o

COMDEL Computes $\Delta a/a$, $\Delta\omega_o$, Δa_{xN}, Δa_{yN}.

REPSUB Uses MNREP to compute Δx, Δy, Δz, $\Delta\dot{x}$, $\Delta\dot{y}$, $\Delta\dot{z}$ at time t.

The diagram in the upper portion of Figure 3-16 illustrates the complete looping sequence of operations between ERP and ERPDIST. This represents the ERPRED function described above.

3.8 PREDICTION DISTRIBUTION - ERPDIST

The Prediction Distribution Program Unit, ERPDIST, performs the statistical processing of the delta position and velocity data generated by the ERP Program Unit. The delta position and velocity data generated by ERP are the differences in the predicted position and velocity vectors produced by the variant element sets and the reference element set. These differences constitute the sets of position and velocity vector errors arising from the simulated observational inaccuracies.

The purpose of the ERPDIST Unit is to compute at each prediction point a statistic of each coordinate of position and velocity. The statistic of the errors include the maximum, minimum, mean, standard deviation, and at option a frequency distribution of the errors. Also, for all prediction points in a quadrant (beginning at the last observation), the statistic of each coordinate at each prediction point will be treated as a set or error quantities, and for each of these corresponding error quantities, a statistic will be computed for that quadrant of the orbit. The process is repeated for the remaining quadrants.

3.8.1 ERPDIST INPUT-OUTPUT DATA

The statistical processing of position and velocity errors require the following input data:

(1) DPVBLOC, the k sets of delta position and velocity error data:

Δx or D_u	$\Delta \dot{x}$ or $\dot{D}_u$
Δy or D_v	$\Delta \dot{y}$ or $\dot{D}_v$
Δz or D_w	$\Delta \dot{z}$ or $\dot{D}_w$
$\lvert \Delta \underline{r} \rvert$	$\lvert \Delta \dot{\underline{r}} \rvert$

These error quantities are output from the Error Prediction Program Unit and are either in equatorial or orbit plane coordinates respectively.

(2) ERPNS The number of samples of a position-velocity type error used to compute the mean and standard deviati on.

(3) QUNS The number of samples of the error quantity used to compute the mean and standard deviation in the quandrant and analysis.

(4) ERPI The number of cells in the position-velocity coordinate frequency distributions.

(5) ERPS1 The number of standard deviations below the mean at the first cell of the position-velocity coordinate frequency distribution.

(6) ERPS2 The number of standard deviations above the mean at the last cell of the position-velocity coordinate frequency distribution.

(7) OTPFLAG Control parameter from ERP defining either equatorial or orbit plane coordinates in DPVBLOC.

(8) CASETYP Control parameter from ERP indicating the option to perform or exclude quadrant analysis.

(9) CAPPI Number of prediction points per quadrant of the orbit.

(10) OPOUTI Control parameter indicating option to include or exclude frequency distribution of position and velocity errors in printed output.

(11) OPOUT2 Control parameter indicating option to include or exclude punch card output.

(12) ERPDOPT Control parameter indicating whether distributions are desired.

Items other than DPVBLOC are input via the ERPRED Control Card.

The output from ERPDIST routine consists of the following data:

(a) Identification of error quantity

(b) Mean error

(c) Standard deviation of error

(d) Minimum error

(e) Maximum error

(f) Time since epoch in minutes

Additional output data, optional to the ERPDIST routine, includes:

(g) Frequency distribution of errors

(h) Punch cards containing items (a) through (f) above

3.8.2 ERPDIST FORMULATION

The presentation of the computations involved in the ERPDIST Unit is given under Common Subroutine, STOB.

3.8.3 ERPDIST PROGRAM SPECIFICATION SUMMARY

a. System Input

DPVBLOC
ERPS1
ERPS2
ERPNS
ERPI
OTPFLG
QUNS
CAPPI
K
CASETYP
QUI
OPOUT1
OPOUT2
ERPDOPT

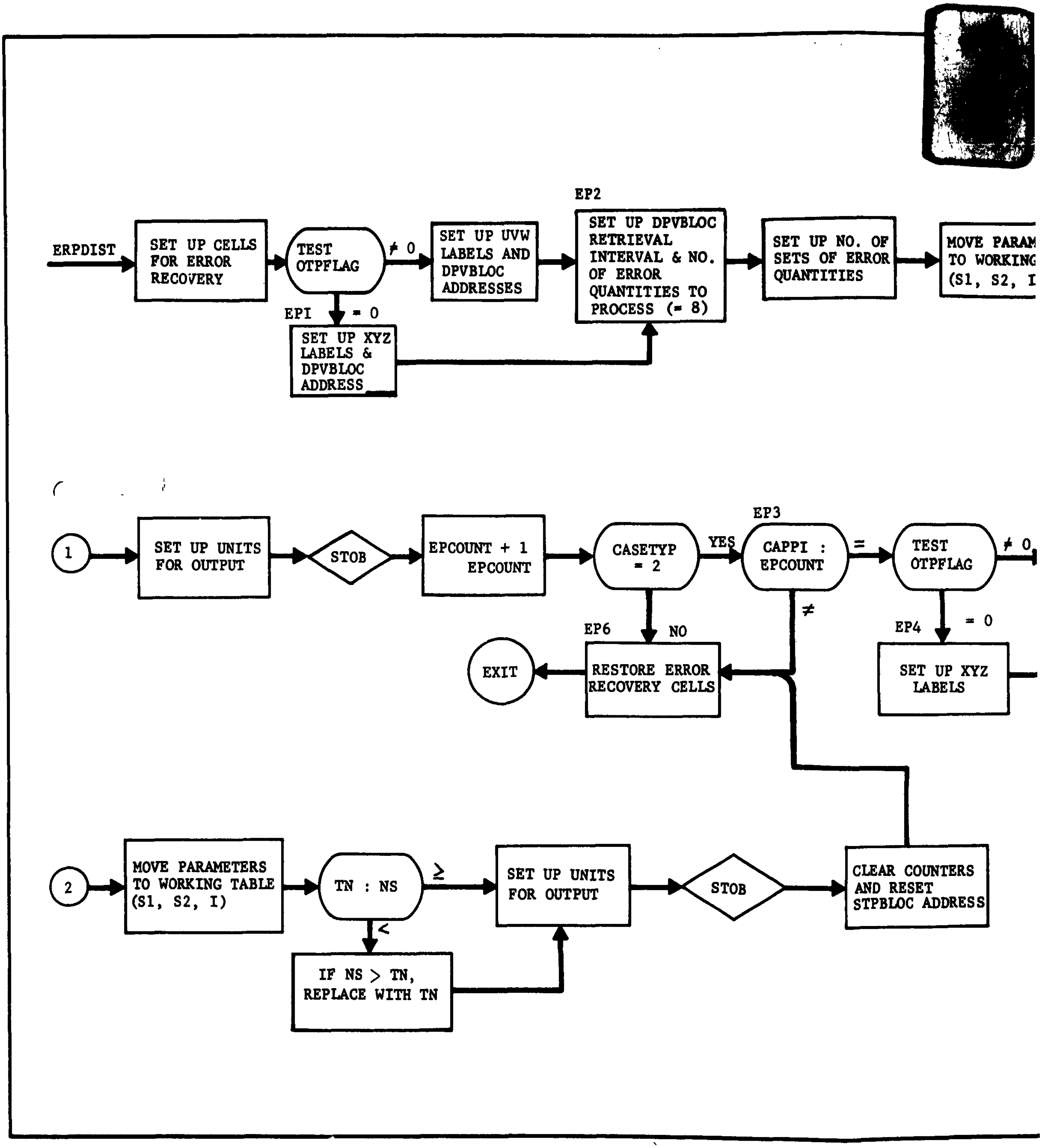
b. Associated Input Cards

ERPRED Control

c. System Output

For each of the position velocity errors (either equatorial or orbit coordinates), the following output is produced:

MU
SIGMA
MIN
MAX
Time since epoch
Frequency Dist



ERPDIST
SET UP CELLS FOR ERROR RECOVERY
TEST OTPFLAG
≠ 0
= 0
EP1
SET UP XYZ LABELS & DPVBLOC ADDRESS
SET UP UVW LABELS AND DPVBLOC ADDRESSES
EP2
SET UP DPVBLOC RETRIEVAL INTERVAL & NO. OF ERROR QUANTITIES TO PROCESS (= 8)
SET UP NO. OF SETS OF ERROR QUANTITIES
MOVE PARAM
TO WORKING
(S1, S2, I
1
SET UP UNITS FOR OUTPUT
STOB
EPCOUNT + 1 EPCOUNT
CASETYP = 2
YES
NO
EP3
CAPPI : EPCOUNT
=
≠
TEST OTPFLAG
≠ 0
= 0
EP4
SET UP XYZ LABELS
EP6
RESTORE ERROR RECOVERY CELLS
EXIT
2
MOVE PARAMETERS TO WORKING TABLE (S1, S2, I)
TN : NS
≥
<
IF NS > TN, REPLACE WITH TN
SET UP UNITS FOR OUTPUT
STOB
CLEAR COUNTERS AND RESET STPBLOC ADDRESS

FIGURE 3-17.

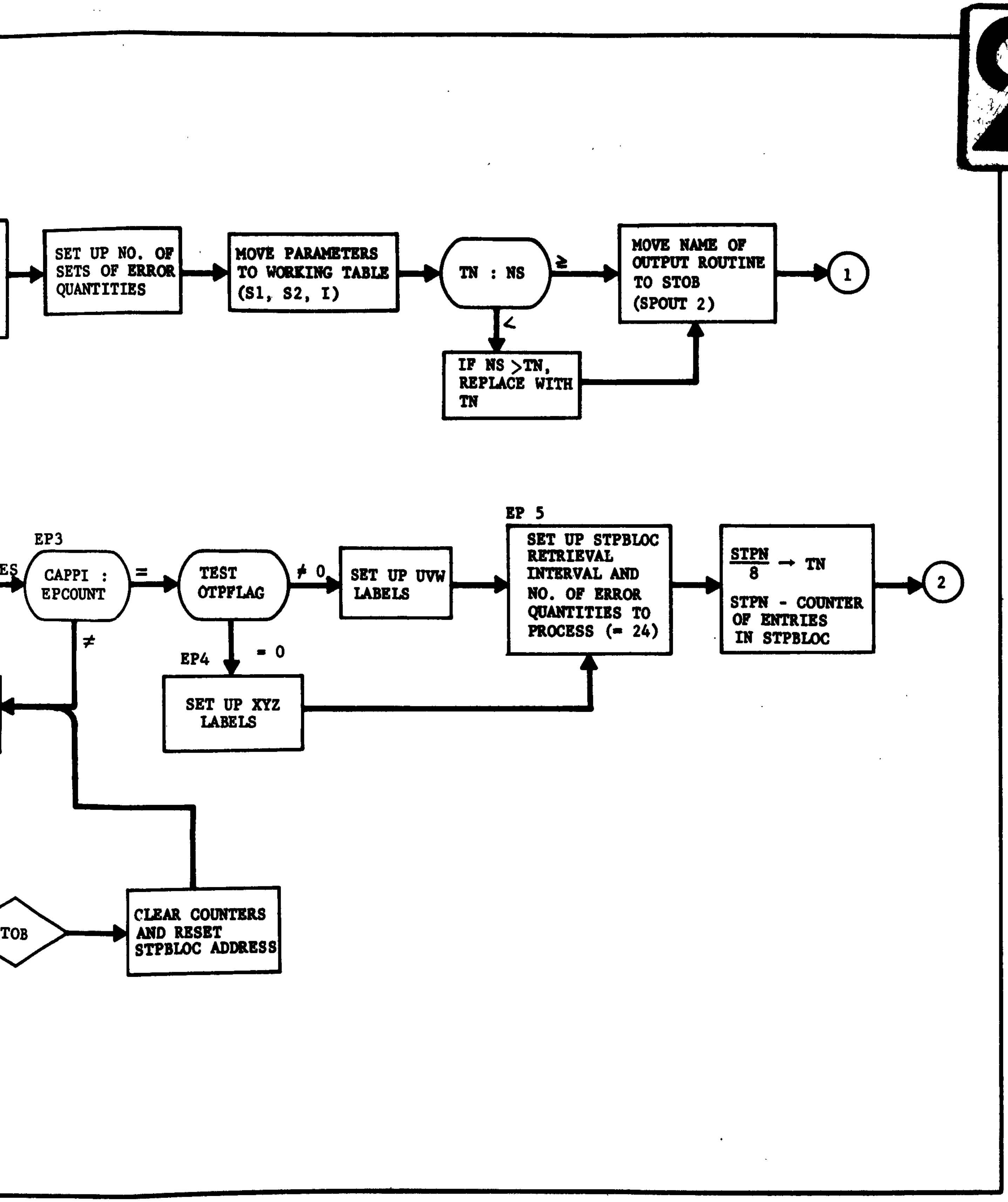

SET UP NO. OF SETS OF ERROR QUANTITIES
MOVE PARAMETERS TO WORKING TABLE (S1, S2, I)
TN : NS
≥
MOVE NAME OF OUTPUT ROUTINE TO STOB (SPOUT 2)
1
<
IF NS >TN, REPLACE WITH TN
EP3
ES
CAPPI : EPCOUNT
=
TEST OTPFLAG
≠ 0
SET UP UVW LABELS
EP 5
SET UP STPBLOC RETRIEVAL INTERVAL AND NO. OF ERROR QUANTITIES TO PROCESS (= 24)
$\frac{\text{STPN}}{8}$ → TN
STPN - COUNTER OF ENTRIES IN STPBLOC
2
≠
EP4
= 0
SET UP XYZ LABELS
TOB
CLEAR COUNTERS AND RESET STPBLOC ADDRESS

FIGURE 3-17. ERPDIST

d. System Common Subroutines

STOB SPOUT2 PANT

e. Philco Subroutines

(None)

3.8.4 ERPDIST SCHEMATIC COMPUTATIONAL SEQUENCE

Figure 3-17 presents the flow diagram for the ERPDIST unit.

SECTION 4

EXECUTIVE PROGRAM

The Executive Program (EP) has two main functions to fulfill in the system: the first, is that of processing all input data and converting them to convenient formats for internal use; the second, is that of providing linkage between the component programs in the system. In order to minimize the size and complexity of the SEA Executive Program, SYS (Philco 2000 Operating System) service routines and subroutines were used whenever possible.

4.1 PROGRAM DESIGN

The SEA EP is divided into three main functional units, one of which contains two program units. These three units accomplish the tasks of input processing and internal program linkage as follows:

(1) General Executive Program (EXEC)

EXEC is the main executive unit which does the basic input processing and linkage.

(2) Special Executive Programs (INTERFN and CONTROLN)

The special executive logic necessary to control a sequence number "N" is contained in CONTROLN and INTERFN. CONTROLN contains input tables, program unit table, etc., and INTERFN contains the executive programming that is actually performed as opposed to control cells and tables. This additional executive control is necessary to implement the special logic required for the specific sequence operations.

(3) Subroutines (SUBS)

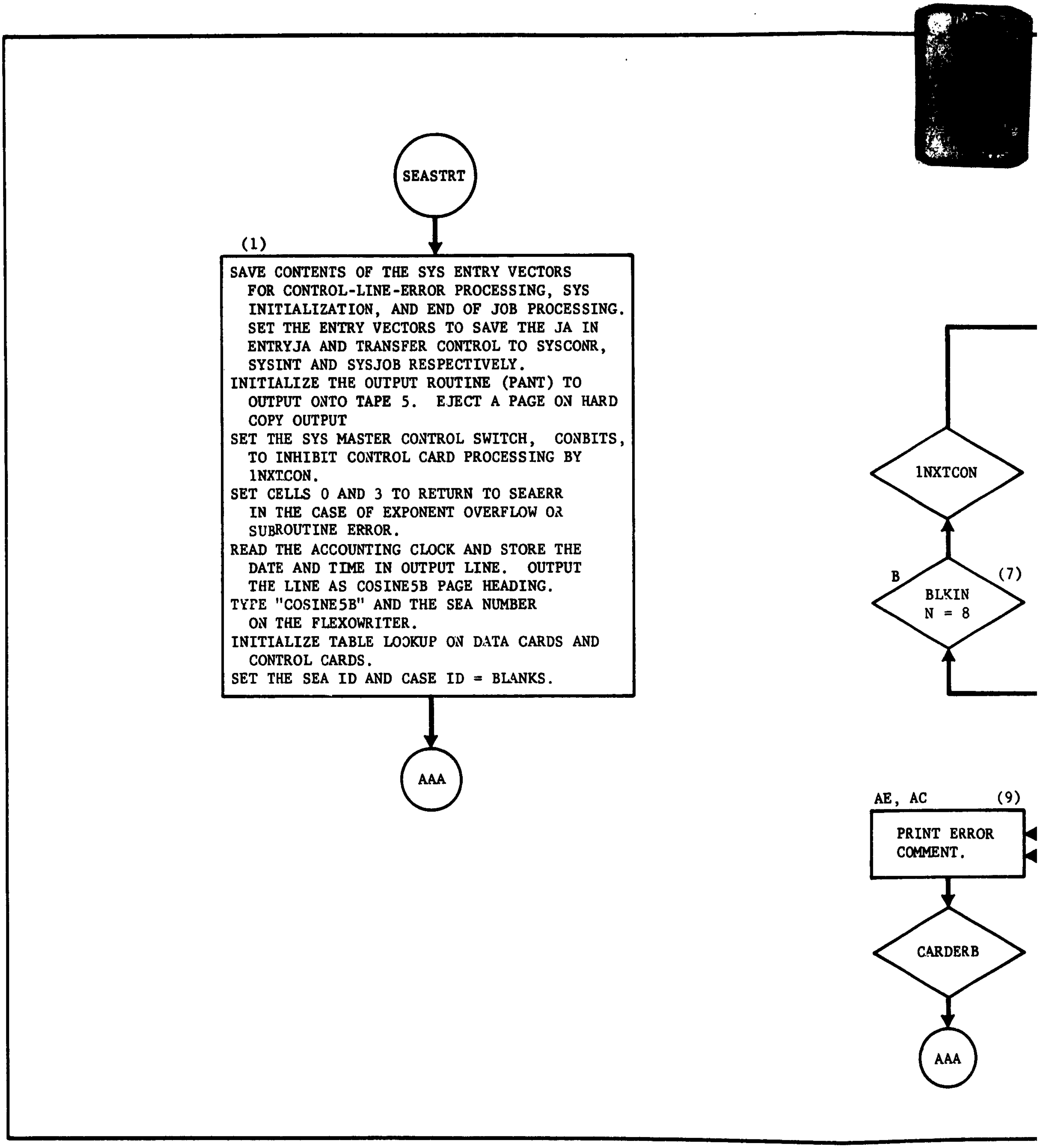
SUBS contains most of the common subroutines used by the SEA program units. SUBS contains card reading routines, retrieval routines, computational subroutines, and other miscellaneous subroutines. A list of all the subroutines is given in the System Subroutine Section.

These four program units (EXEC, CONTROLN, INTERFN, and SUBS) are necessary in all SEA sequences. Therefore, they are loaded along with the specific program units required for each sequence.

4.2 DESCRIPTION OF GENERAL EXECUTIVE LOGIC

The flow diagram of the general logic of the Executive Program is presented in Figures 4-1 through 4-4. In order to associate the description of the executive logic with the schematic representation, various points in the flow diagram are designated numerically (n) and are referenced in this manner in the text. After SYS has brought an SEA sequence into core memory, control is transferred to the EP at SEASTRT (1). This routine does all the initialization of the EP, such as clearing switches and indicators, setting up error recoveries, initializing the output routine, and outputing headings. After accomplishing its tasks, SEASTRT transfers control to AAA (the central input processor). AAA retrieves an input card using 1NXTCON (2) and examines columns 17 to 24 and 80 of each input card to determine its function (3), (4), (5). If it is a legal SEA card, SYS card, or data card, AAA jumps to the correct processing routine or subroutine (6), (7), (8). If the card is not a legal type, AAA rejects the card with a diagnostic comment (9). Most of the card processing routines return to AAA. CONINT is the control card process (18) and always returns to AAA. Routines that do not always return to AAA are SYSRET, JOB, SEACARD and ENDDATA.

A "SYS"card or a "JOB" card will signal SEA to return to SYS. SYSRET and JOB (20), (26) return control to SYS after it has been restored (24), (25). A case ("SEA") card causes the EP to initialize for a case. This is done in SEACARD (11). SEACARD can transfer control to NXJOB if the case card is illegal (17) or to ENDDATA if two case cards are found with no ENDDATA card separating them (16), otherwise, SEACARD returns to AAA (14), (15). An "ENDDATA" card signals SEA to run the case just input. The ENDDATA routine checks the input received against the required input to run (Boxes 12 and 13). ENDDATA never returns to AAA. If the case had any serious input errors, ENDDATA will transfer control to



SAVE CONTENTS OF THE SYS ENTRY VECTORS FOR CONTROL-LINE-ERROR PROCESSING, SYS INITIALIZATION, AND END OF JOB PROCESSING. SET THE ENTRY VECTORS TO SAVE THE JA IN ENTRYJA AND TRANSFER CONTROL TO SYSCONR, SYSINT AND SYSJOB RESPECTIVELY.

INITIALIZE THE OUTPUT ROUTINE (PANT) TO OUTPUT ONTO TAPE 5. EJECT A PAGE ON HARD COPY OUTPUT

SET THE SYS MASTER CONTROL SWITCH, CONBITS, TO INHIBIT CONTROL CARD PROCESSING BY 1NXTCON.

SET CELLS 0 AND 3 TO RETURN TO SEAERR IN THE CASE OF EXPONENT OVERFLOW OR SUBROUTINE ERROR.

READ THE ACCOUNTING CLOCK AND STORE THE DATE AND TIME IN OUTPUT LINE. OUTPUT THE LINE AS COSINE5B PAGE HEADING.

TYPE "COSINE5B" AND THE SEA NUMBER ON THE FLEXOWRITER.

INITIALIZE TABLE LOOKUP ON DATA CARDS AND CONTROL CARDS.

SET THE SEA ID AND CASE ID = BLANKS.

PRINT ERROR COMMENT.

FIGURE 4-1. E

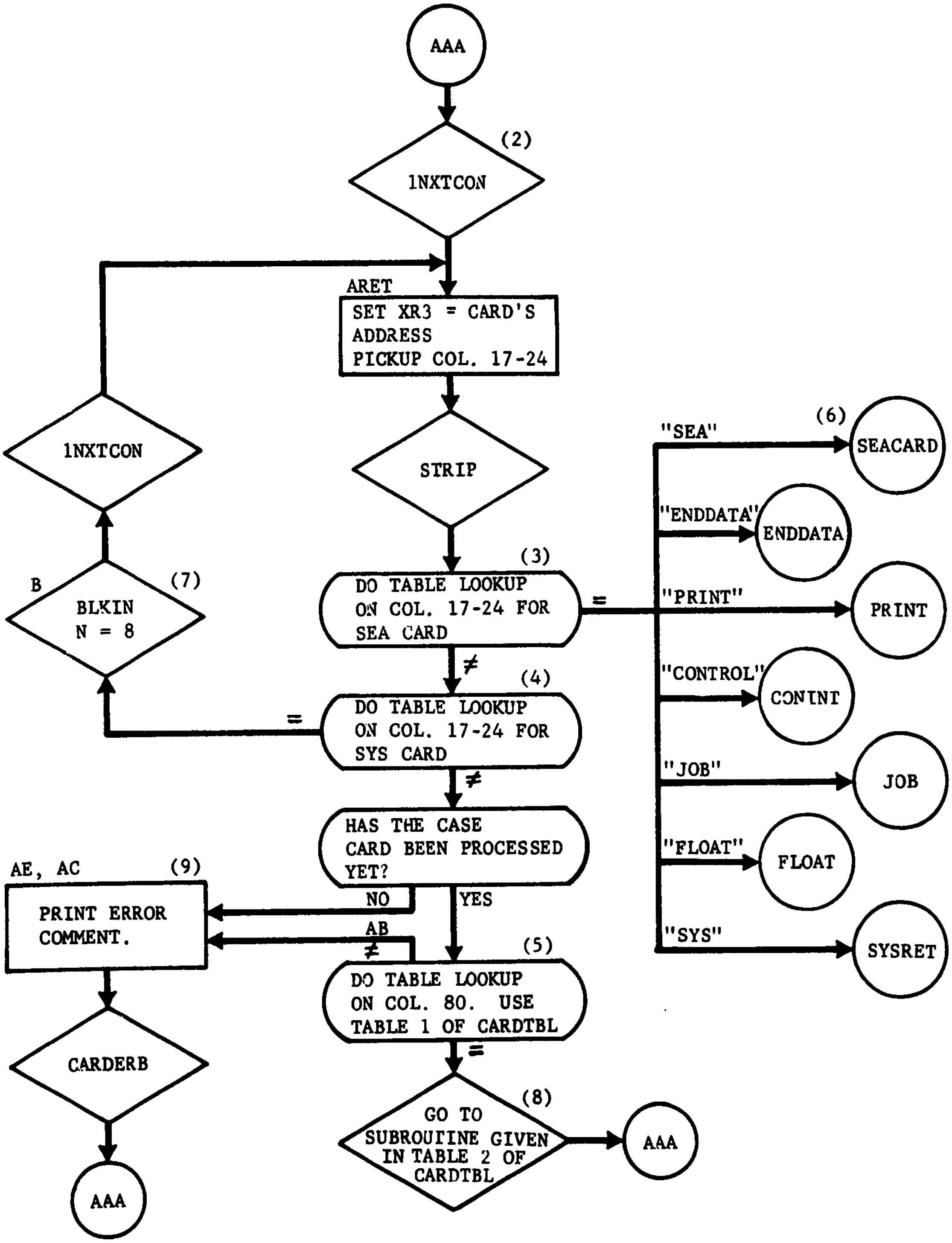

AAA
(2)
1NXTCON
ARET
SET XR3 = CARD'S ADDRESS PICKUP COL. 17-24
STRIP
1NXTCON
B
(7)
BLKIN N = 8
(3)
DO TABLE LOOKUP ON COL. 17-24 FOR SEA CARD
=
≠
(4)
DO TABLE LOOKUP ON COL. 17-24 FOR SYS CARD
=
≠
HAS THE CASE CARD BEEN PROCESSED YET?
NO
YES
AE, AC
(9)
PRINT ERROR COMMENT.
AB
≠
(5)
DO TABLE LOOKUP ON COL. 80. USE TABLE 1 OF CARDTBL
=
CARDERB
AAA
(8)
GO TO SUBROUTINE GIVEN IN TABLE 2 OF CARDTBL
AAA
(6)
"SEA"
SEACARD
"ENDDATA"
ENDDATA
"PRINT"
PRINT
"CONTROL"
CONINT
"JOB"
JOB
"FLOAT"
FLOAT
"SYS"
SYSRET

FIGURE 4-1. EXECUTIVE PROGRAM

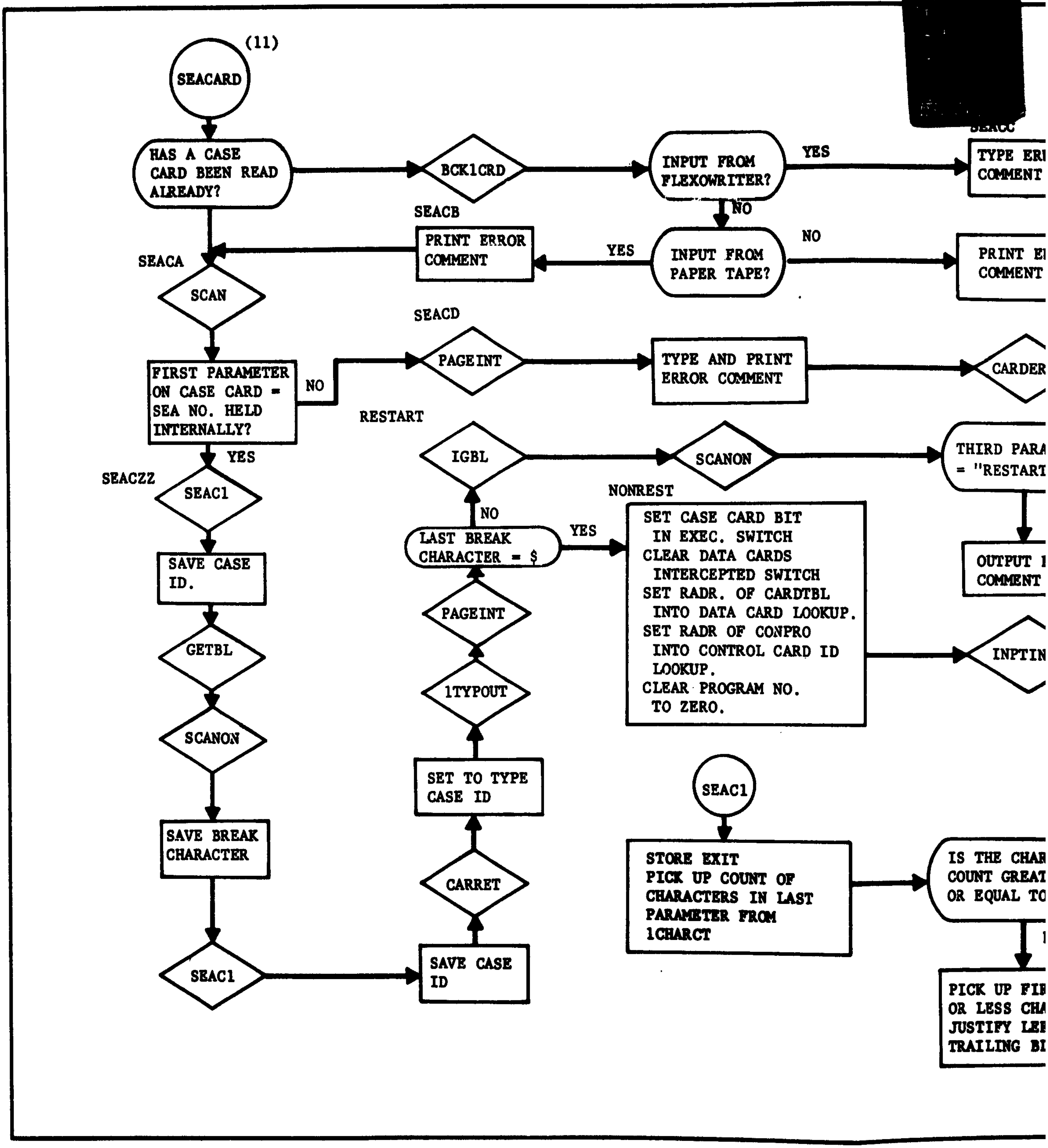

(11)
SEACARD
HAS A CASE CARD BEEN READ ALREADY?
BCK1CRD
INPUT FROM FLEXOWRITER?
YES
TYPE ERI
COMMENT
SEACC
[redacted]
NO
SEACB
PRINT ERROR COMMENT
YES
INPUT FROM PAPER TAPE?
NO
PRINT EI
COMMENT
SEACA
SCAN
SEACD
PAGEINT
TYPE AND PRINT ERROR COMMENT
CARDER
FIRST PARAMETER ON CASE CARD = SEA NO. HELD INTERNALLY?
NO
RESTART
YES
IGBL
SCANON
THIRD PARA
= "RESTART
SEACZZ
SEAC1
NONREST
NO
LAST BREAK CHARACTER = $
YES
SET CASE CARD BIT IN EXEC. SWITCH
CLEAR DATA CARDS INTERCEPTED SWITCH
SET RADR. OF CARDTBL INTO DATA CARD LOOKUP.
SET RADR OF CONPRO INTO CONTROL CARD ID LOOKUP.
CLEAR PROGRAM NO. TO ZERO.
OUTPUT I
COMMENT
SAVE CASE ID.
GETBL
PAGEINT
INPTIN
1TYPOUT
SCANON
SET TO TYPE CASE ID
SEAC1
SAVE BREAK CHARACTER
CARRET
STORE EXIT
PICK UP COUNT OF CHARACTERS IN LAST PARAMETER FROM 1CHARCT
IS THE CHAR
COUNT GREAT
OR EQUAL TO
SEAC1
SAVE CASE ID
PICK UP FII
OR LESS CHA
JUSTIFY LEI
TRAILING BI

FIGURE 4-

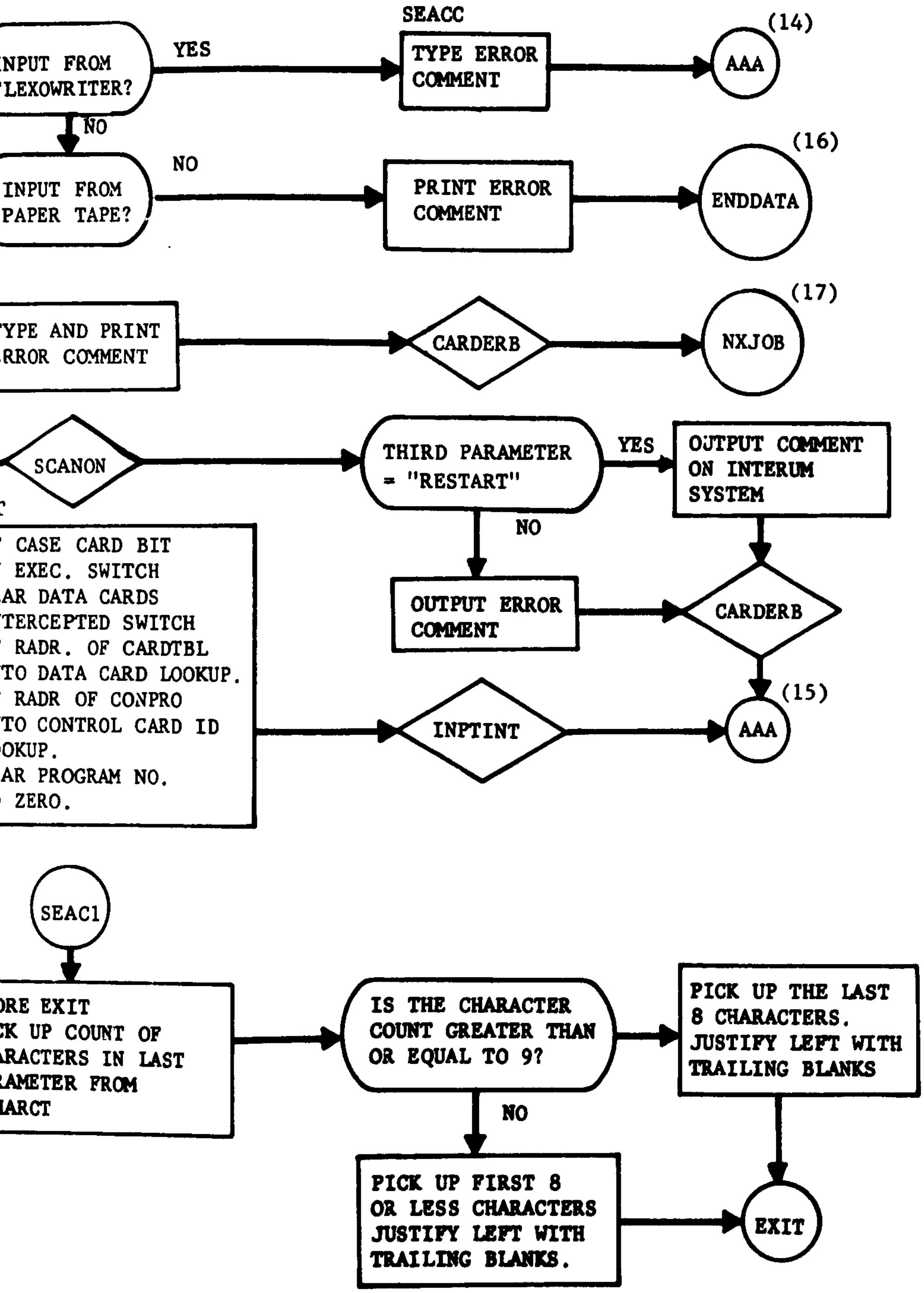


FIGURE 4-2. EXECUTIVE PROGRAM (Continued)

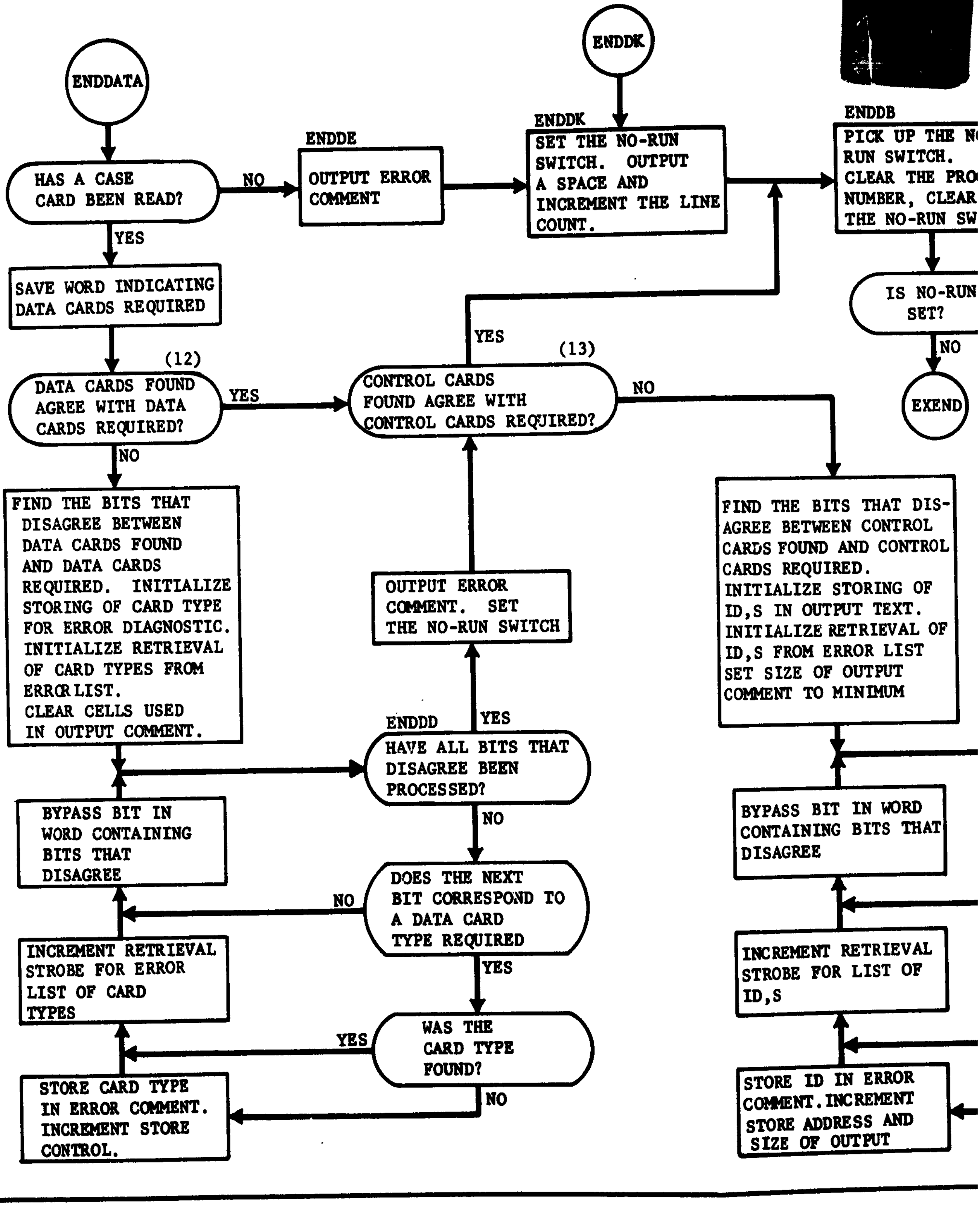

ENDDATA
HAS A CASE CARD BEEN READ?
NO
YES
ENDDE
OUTPUT ERROR COMMENT
ENDDK
ENDDK
SET THE NO-RUN SWITCH. OUTPUT A SPACE AND INCREMENT THE LINE COUNT.
ENDDB
PICK UP THE N
RUN SWITCH.
CLEAR THE PRO
NUMBER, CLEAR
THE NO-RUN SW
IS NO-RUN SET?
NO
EXEND
SAVE WORD INDICATING DATA CARDS REQUIRED
(12)
DATA CARDS FOUND AGREE WITH DATA CARDS REQUIRED?
YES
NO
(13)
CONTROL CARDS FOUND AGREE WITH CONTROL CARDS REQUIRED?
YES
NO
FIND THE BITS THAT DISAGREE BETWEEN DATA CARDS FOUND AND DATA CARDS REQUIRED. INITIALIZE STORING OF CARD TYPE FOR ERROR DIAGNOSTIC. INITIALIZE RETRIEVAL OF CARD TYPES FROM ERRORLIST. CLEAR CELLS USED IN OUTPUT COMMENT.
OUTPUT ERROR COMMENT. SET THE NO-RUN SWITCH
FIND THE BITS THAT DIS-AGREE BETWEEN CONTROL CARDS FOUND AND CONTROL CARDS REQUIRED. INITIALIZE STORING OF ID,S IN OUTPUT TEXT. INITIALIZE RETRIEVAL OF ID,S FROM ERROR LIST SET SIZE OF OUTPUT COMMENT TO MINIMUM
ENDDD
YES
HAVE ALL BITS THAT DISAGREE BEEN PROCESSED?
NO
BYPASS BIT IN WORD CONTAINING BITS THAT DISAGREE
BYPASS BIT IN WORD CONTAINING BITS THAT DISAGREE
DOES THE NEXT BIT CORRESPOND TO A DATA CARD TYPE REQUIRED
NO
YES
INCREMENT RETRIEVAL STROBE FOR ERROR LIST OF CARD TYPES
INCREMENT RETRIEVAL STROBE FOR LIST OF ID,S
WAS THE CARD TYPE FOUND?
YES
NO
STORE CARD TYPE IN ERROR COMMENT. INCREMENT STORE CONTROL.
STORE ID IN ERROR COMMENT. INCREMENT STORE ADDRESS AND SIZE OF OUTPUT

FIGURE 4-3. EXE

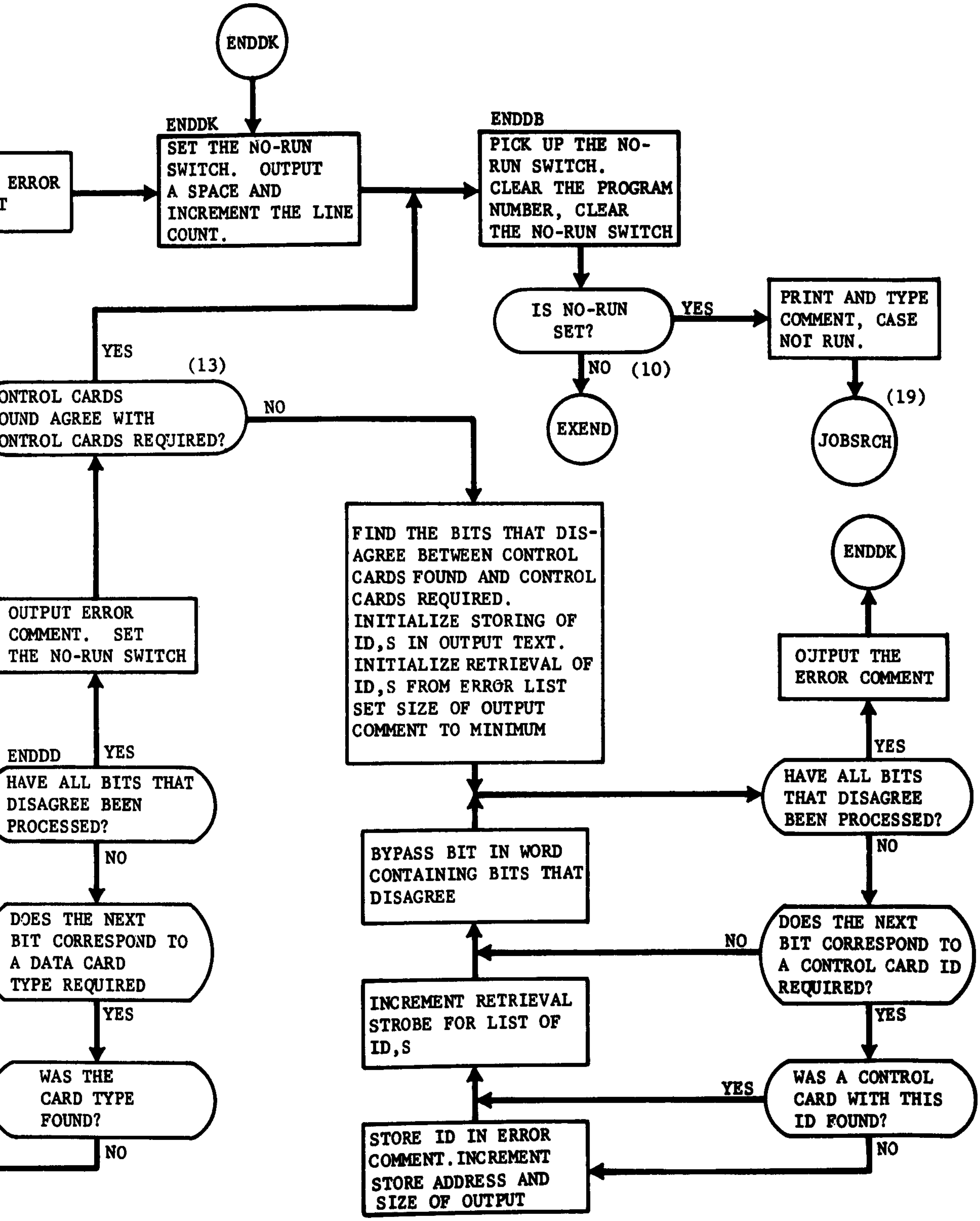
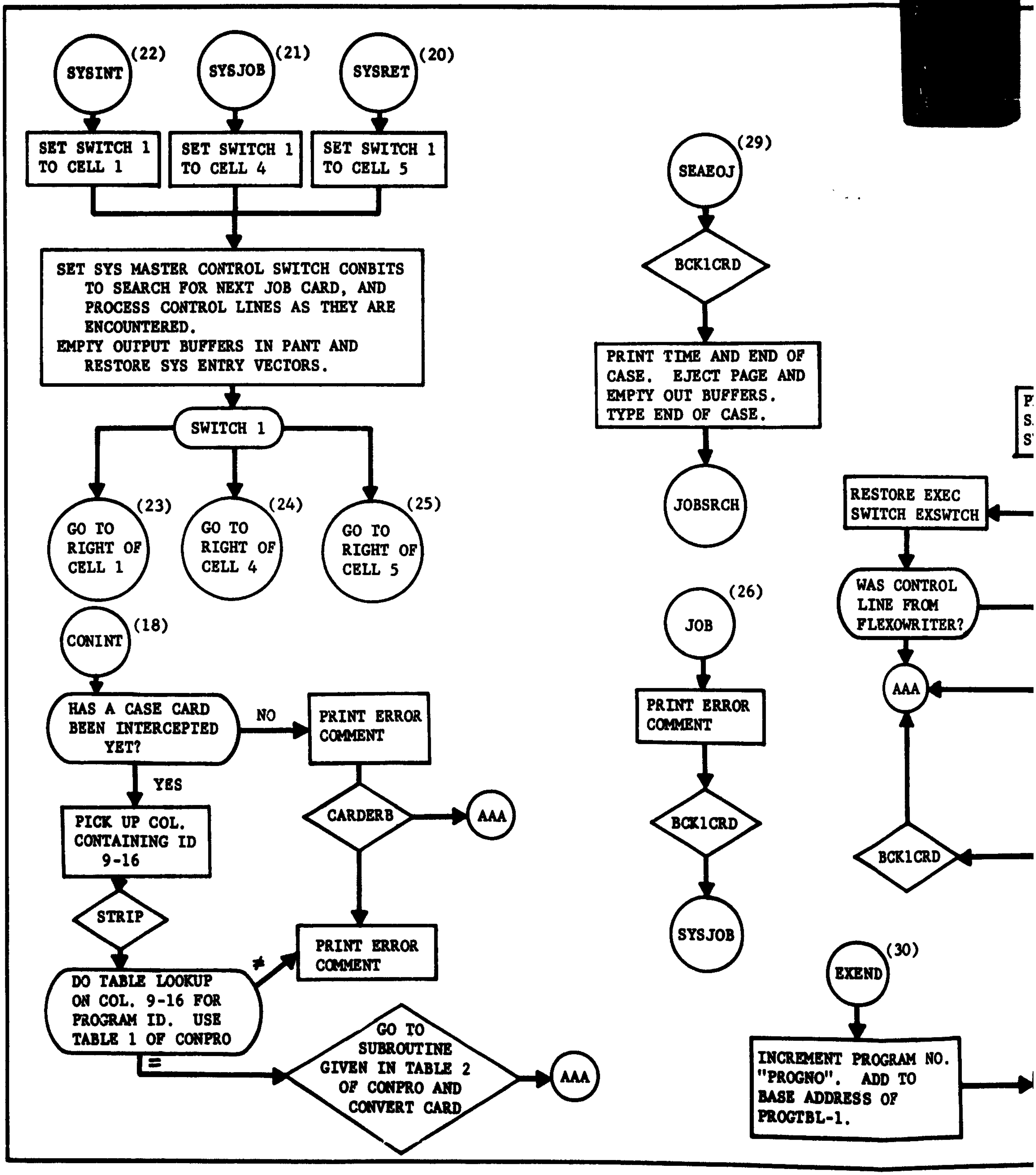

ENDDK
ENDDK
SET THE NO-RUN SWITCH. OUTPUT A SPACE AND INCREMENT THE LINE COUNT.
ERROR
T
ENDDB
PICK UP THE NO-RUN SWITCH. CLEAR THE PROGRAM NUMBER, CLEAR THE NO-RUN SWITCH
IS NO-RUN SET?
YES
PRINT AND TYPE COMMENT, CASE NOT RUN.
NO
(10)
EXEND
(19)
JOBSRCH
YES
(13)
ONTROL CARDS OUND AGREE WITH ONTROL CARDS REQUIRED?
NO
FIND THE BITS THAT DISAGREE BETWEEN CONTROL CARDS FOUND AND CONTROL CARDS REQUIRED. INITIALIZE STORING OF ID,S IN OUTPUT TEXT. INITIALIZE RETRIEVAL OF ID,S FROM ERROR LIST SET SIZE OF OUTPUT COMMENT TO MINIMUM
ENDDK
OUTPUT ERROR COMMENT. SET THE NO-RUN SWITCH
OUTPUT THE ERROR COMMENT
ENDDD
YES
HAVE ALL BITS THAT DISAGREE BEEN PROCESSED?
YES
HAVE ALL BITS THAT DISAGREE BEEN PROCESSED?
NO
NO
BYPASS BIT IN WORD CONTAINING BITS THAT DISAGREE
DOES THE NEXT BIT CORRESPOND TO A DATA CARD TYPE REQUIRED
NO
DOES THE NEXT BIT CORRESPOND TO A CONTROL CARD ID REQUIRED?
YES
INCREMENT RETRIEVAL STROBE FOR LIST OF ID,S
YES
WAS THE CARD TYPE FOUND?
YES
WAS A CONTROL CARD WITH THIS ID FOUND?
NO
NO
STORE ID IN ERROR COMMENT. INCREMENT STORE ADDRESS AND SIZE OF OUTPUT

FIGURE 4-3. EXECUTIVE PROGRAM (Continued)


SYSINT (22)
SYSJOB (21)
SYSRET (20)
SET SWITCH 1 TO CELL 1
SET SWITCH 1 TO CELL 4
SET SWITCH 1 TO CELL 5
SET SYS MASTER CONTROL SWITCH CONBITS TO SEARCH FOR NEXT JOB CARD, AND PROCESS CONTROL LINES AS THEY ARE ENCOUNTERED.
EMPTY OUTPUT BUFFERS IN PANT AND RESTORE SYS ENTRY VECTORS.
SWITCH 1
GO TO RIGHT OF CELL 1 (23)
GO TO RIGHT OF CELL 4 (24)
GO TO RIGHT OF CELL 5 (25)
CONINT (18)
HAS A CASE CARD BEEN INTERCEPTED YET?
NO
PRINT ERROR COMMENT
YES
PICK UP COL. CONTAINING ID 9-16
CARDERB
AAA
STRIP
PRINT ERROR COMMENT
≠
DO TABLE LOOKUP ON COL. 9-16 FOR PROGRAM ID. USE TABLE 1 OF CONPRO
=
GO TO SUBROUTINE GIVEN IN TABLE 2 OF CONPRO AND CONVERT CARD
AAA
SEAEOJ (29)
BCK1CRD
PRINT TIME AND END OF CASE. EJECT PAGE AND EMPTY OUT BUFFERS. TYPE END OF CASE.
JOBSRCH
JOB (26)
PRINT ERROR COMMENT
BCK1CRD
SYSJOB
P
S
S
RESTORE EXEC SWITCH EXSWTCH
WAS CONTROL LINE FROM FLEXOWRITER?
AAA
BCK1CRD
EXEND (30)
INCREMENT PROGRAM NO. "PROGNO". ADD TO BASE ADDRESS OF PROGTBL-1.

FIGURE

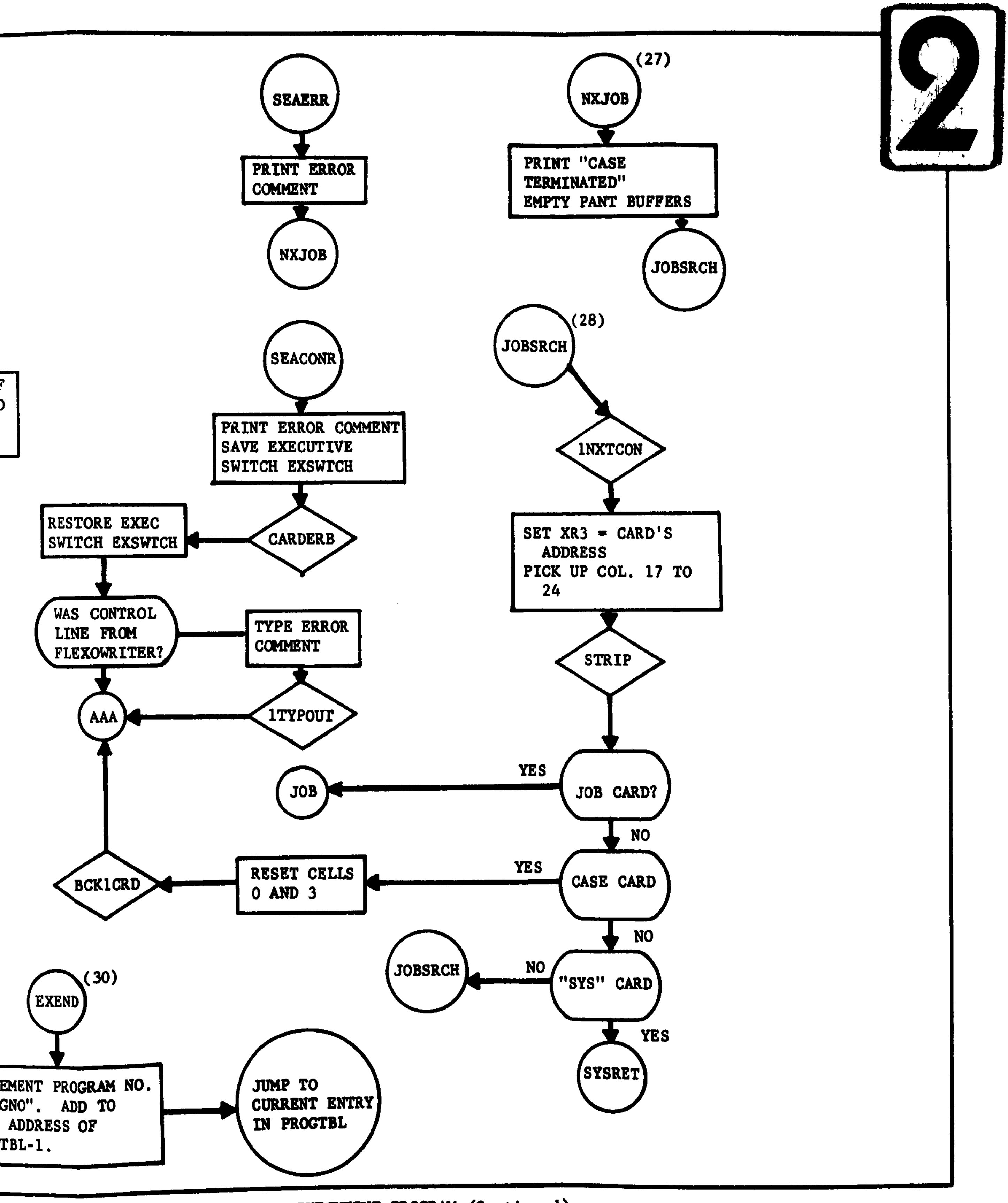

2
SEAERR
PRINT ERROR COMMENT
NXJOB
(27)
NXJOB
PRINT "CASE TERMINATED" EMPTY PANT BUFFERS
JOBSRCH
SEACONR
PRINT ERROR COMMENT SAVE EXECUTIVE SWITCH EXSWTCH
(28)
JOBSRCH
1NXTCON
F
D
RESTORE EXEC SWITCH EXSWTCH
CARDERB
SET XR3 = CARD'S ADDRESS PICK UP COL. 17 TO 24
WAS CONTROL LINE FROM FLEXOWRITER?
TYPE ERROR COMMENT
STRIP
AAA
1TYPOUT
YES
JOB
JOB CARD?
NO
YES
BCK1CRD
RESET CELLS 0 AND 3
CASE CARD
NO
JOBSRCH
NO
"SYS" CARD
(30)
EXEND
YES
SYSRET
EMENT PROGRAM NO. GNO". ADD TO ADDRESS OF TBL-1.
JUMP TO CURRENT ENTRY IN PROGTBL

FIGURE 4-4. EXECUTIVE PROGRAM (Continued)

JOBSRCH (19) initiating a search for the next case or job. If the case did not have serious input errors, ENDDATA will exit to EXEND (10) and initiate the running of the case. "PRINT" and "FLOAT" cards will execute the PRINT or FLOAT subroutines described in the System Subroutine Specifications.

The central routine used in sequence control is EXEND (30). EXEND increments the program number (PROGNO) and transfers control to the next program in sequence, using PROGTBL. Normally, programs return to EXEND; however, if a nonrecoverable error occurs during a case run, the program unit puts out a diagnostic comment and exits to NXJOB (27). After putting out a comment indicating that the case was terminated, NXJOB transfers control to JOBSRCH (28).

Normal termination of a case is executed by a jump to SEAEOJ (29). This routine is the last entry in PROGTBL and the EXEND logic will result in a transfer of control to SEAEOJ after completion of the last program unit. SEAEOJ puts out a comment indicating that the case has ended and transfers control to JOBSRCH.

JOBSRCH (28) searches the input until a SYS, JOB, or case card is found, in which case it transfers control to SYSRET (20), JOB (26), or AAA (2), respectively.

The SEA system will retain control until a SYS or JOB card is intercepted, or until the operator executes a jump to cell 1, 4, or 5 from the console in an attempt to initialize SYS or run the next job. In any of these conditions, the EP will restore the SYS entry vectors, reset the SYS master control switch CONBITS, and exit to the right half of cell 1 (23), 4 (24), or 5 (25). The exit taken depends on the next SYS function desired.

4.3 DESCRIPTION OF SPECIAL EXECUTIVE LOGIC

Situations arise in any system that require special logic and handling. Rather than incoporate these in a general EP, the SEA system isolates these "special executive" routines in the program unit INTERFN. These perform the logic needed during a case run to accomplish special tasks required, such as performing a Monte Carlo-Differential Correction loop. These are written as small sequence programs and have entries in PROGTBL. They therefore return to EXEND. At present, the special executive routines in the SEA system and their functions are as follows:

EXEC1AA	Outputs nominal elements. Restricts sensor data to one sensor in Type C and D elements. EXEC1AA is the first entry in PROGTBL.

EXEC1B Initializes Differential Correction and Monte Carlo.
Initializes MOCARO - SGPDC loop. EXEC1B is between OBSIM and MOCARO in the PROGTBL.

EXEC1C Increments counter of MOCARO - SGPDC loops (KCOUNT) and compares it to the number of runs requested on the case input (K). If more loops are required, EXEC1C decrements PROGNO by 3 and exits to EXEND. If not, EXEC1C outputs the final random number and number of unsuccessful corrections and then exits to EXEND without altering PROGNO. EXEC1C is between SGPDC and DCDIST in the PROGTBL.

4.4 DESCRIPTION OF EXECUTIVE ROUTINES AND SUBROUTINES

EXEND Normal end of program return.

(EXEC) If a sequence program is executed in a normal fashion, i.e., no fatal errors detected, the program returns to EXEND signalling the EP to transfer control to the next program in sequence. EXEND increments the left address of PROGNO by 1 and adds it to the address of PROGTBL-1. This is the location of a jump to the next program in the sequence.

NXJOB or NXTJOB Error end of sequence exit.

(EXEC) If a sequence program encounters a fatal error when attempting to run a case, control is returned to NXJOB, following a diagnostic comment. NXJOB indicates that the case is terminated both on the flexo and on hard copy. It then transfers control to JOBSRCH to begin the next case.

JOBSRCH Job or case search entrance.

(EXEC) JOBSRCH processes the input data on Tape 8, searching for one of the following three cards:

Col 17-24 = SEA - case card
= SYS - SYS card
= JOB - JOB card

If a case card is found, JOBSRCH repositions the input to the case card, resets cells 0 and 3 for the EP error processing, and transfers control to AAA. If a SYS card is found, JOBSRCH transfers control to SYSRET. If a JOB card is found, JOBSRCH transfers control to JOB.

JOB JOB card processing.

(EXEC) JOB types a comment on the flexo indicating that a JOB card was intercepted, repositions the input retrieval to the JOB card and exits to SYSJOB.

SYSJOB SYS job processing.

(EXEC) SYSJOB resets CONBITS, empties the output buffers in PANT, restores the SYS entry vectors, and exits to M/4 right.

SYSRET Return control to SYS

(EXEC) SYSRET resets CONBITS, empties the output buffers in PANT, restores the SYS entry vectors, and exits to M/5 right.

SYSINT Initialize SYS

(EXEC) SYSINT resets CONBITS, empties the output buffers in PANT, restores the SYS entry vectors, and exits to M/1 right.

SEASTRT Initialization of SEA exec.

(EXEC) SEASTRT does all the initialization required of the SEA executive routine. It saves the contents of the SYS entry vectors 1SYSIN, 1ENJOB, and XCONER and changes them to set the ENTRYJA and transfer control to SYSINT, SYSJOB and SEACONR, respectively; initializes PANT to output onto Tape 5, ejects a page, and outputs COSINE 5B headings with the date and time; sets CONBITS for SEA usage; sets cells 0 and 3 to return to SEAERR; sets the size of the data card table and control line ID table = 0 (to prevent table lookup from matching until after a case card); blanks the SEA and case ID's; and then exits to AAA.

SEACONR Control line error processing.

(EXEC) If an error occurs on SYS control lines, SEACONR prints an error comment on hard copy, and types an error comment if input is from the flexo. In either case, control is returned to AAA.

SEAERR Exponent overflow or subroutine error processing.

(EXEC) SEAERR outputs an error comment and then transfers control to NXJOB.

AAA Central Executive input processor

(EXEC) AAA uses 1NXTCON to retrieve the next input card, and uses STRIP to extract the command field (Col 17-24). AAA uses table A1A to determine if the card is an SEA card. If equality is found with one of the SEA card functions, AAA jumps to the corresponding entry in table A1B to convert the card. Subroutines that process SEA cards are SEACARD (Case card), ENDDATA (Enddata card), PRINT (Print card), FLOAT (Float card), CONINT (Control card), JOB (Job card), and SYSRET (SYS card). Subroutines PRINT, FLOAT, and CONINT will return to AAA; JOB and SYSRET restore SYS and return control to SYS; SEACARD normally returns control to AAA, however can go to ENDDATA or NXJOB; and ENDDATA always returns control to EXEND or JOBSRCH.

If the card is not an SEA card, then AAA uses TABLE1 to determine if the card is a legal SYS card. If it is, control is transferred to BLKIN with a parameter of 8 (Block 8 of the SYS Tape desired). BLKIN will locate Block 8 which will trigger the SYS card processing. After the card has been processed, SYS will return to 1NXTCON which will retrieve the next card, and return to ARET (in AAA). If the input card is not an SEA or SYS card, AAA checks to see if a Case card has been found. If not, data cards are not yet allowed, and the card will be rejected with a diagnostic comment. If a Case card has been found, AAA will compare Col 80 of the input card to legal data card types as given in the table specified in CARDTBL left address. If equality is found, AAA jumps to the corresponding entry in the table specified in CARDTBL right address, which is a subroutine to process the particular data card. After the card has been processed, control is returned to AAA. If the input card is none of the above legal inputs, it is rejected with a comment indicating that an error exists in Col 17-24 or 80. Control is then returned to AAA.

SEACARD Case Card Processor

(EXEC) SEACARD processes a Case card (SEA in Col 17-24). It checks the SEA number against the sequence number held internally and sets up the SEA ID and Case ID on the page headings. It also clears all switches and indicators, and initializes the input processing including table lookups on data cards and control card ID's. SEACARD normally exits to AAA, however, with various error conditions, it can exit to ENDDATA (previous Case card encountered with no ENDDATA card on case), or NXJOB (SEA ID disagreement).

ENDDATA ENDDATA Card Processor

(EXEC) ENDDATA processes an ENDDATA card. This primarily consists of checking the data successfully processed in the case and comparing it to the minimal data requirements for the sequence being run. ENDDATA outputs error diagnostics indicating which types of data cards were not found and which types of control cards were not found. Deficiencies in either type of card will cause ENDDATA to inhibit the running of the case. If all data required was found, ENDDATA checks the no-run switch (bit0-EXSWTCH) for an inhibit set during the input processing. If no run is set, ENDDATA will not run the case, but transfers control to JOBSRCH. If no errors were found and the case is to be run, ENDDATA transfers control to EXEND.

CONINT Control Card Processor

(EXEC) CONINT processes CONTROL (CONTROL in Col 17-24) cards. Using the table of N entries (N specified by CONPRO right address) beginning at a location TABLE (specified by CONPRO left address), CONINT does a table lookup on the CONTROL card, ID given in Col 9-16. If equality is found, CONINT jumps to N plus the core location in TABLE equal to the ID. This must be a jump to a subroutine to process the CONTROL card, and a jump to AAA. CONINT therefore returns control to AAA.

The subroutines used by CONINT must set a bit in cell CCARD after a successful card read is executed. This signals ENDDATA that the specific control card was found and read without any errors. For bit assignments, see cell CCARD.

SEAEOJ Sequence Termination

(EXEC) SEAEOJ terminates an SEA sequence. After printing and typing "END OF CASE" SEAEOJ empties the output buffers and exits to JOBSRCH for the next case.

INPTINT Initialization Subroutine

(INTERFN) When the EXEC encounters a Case card, it will go to the subroutine INPTINT. This should do all necessary initialization of the input routines for the sequence, such as setting up strobes and clearing counters for various data blocks (i.e., EBLOC, OBLOC, etc.).

4.5 DESCRIPTION OF EXECUTIVE CONTROL SWITCHES AND TABLES

EXSWTCH bit 0 No-run switch

(EXEC) If = 0, no fatal input errors

If = 1, fatal errors on input, do not run case through sequence

bit 1 Case card intercepted

If = 0, case card not intercepted

If = 1, case card found and processed

bit 2-47 Not used

CARDS Cards found. A bit is set = 1 for each card type intercepted and successfully processed since last SEA card. Bits are assigned as follows:

(EXEC)

bit 0	Not used
1	Element cards (Type E)
2	Sensor cards (Type S)
3	Simulation parameter cards (Type I)
4	Standard deviation and bias cards (Type M)
5	Variant element cards (Type V)
6-45	Not used
46	Observation cards (Type ϕ, △, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
47	Not used

CARDOPT (CONTROLN) Cards required. A bit is set = 1 for each card type required to run the sequence. Bit assignments must be identical to the bit assignments for CARDS.

CRDETYP (CONTROLN) C/HLT, TABLE; C/HLT, where location TABLE must contain the following:

```
TABLE  W/0000000A               Where A, B, C, etc.
       W/0000000B      etc.     correspond to the
       W/0000000C               bits in CARDOPT.
```

The first entry in TABLE must agree with the leftmost bit set = 1 in CARDOPT, the second entry with the second bit = 1, etc. There must be the same number of entries in TABLE as there are bits = 1 in CARDOPT.

CCARD (EXEC) Control cards found. A bit is set = 1 for each type of control card successfully processed since last SEA card. Bits are assigned as follows:

bit 0	Not used
1	AQTINT control card
2	MOCARO control card
3	SGPDC control card
4	Not used
5	ERPRED control card
6-47	Not used

CCRDOPT Control cards required. A bit is set = 1 for each type of control card required to run the sequence.

(CONTROLN) Bit assignments must be the same as those for CCARD.

CONPROZ C/HLT; C/HLT, TABLE where location TABLE must contain the following:

(CONTROLN)

```
TABLE   W/AQTINT
        W/MOCARO
        W/SGPDC, etc.
```

The BCD words given correspond to the ID's expected on the control cards required as specified by CCRDOPT. The first entry in TABLE will correspond to the first bit (counting from the left) in CCRDOPT. The number of entries in TABLE will be the same as the number of bits set in CCRDOPT. Each entry should be centered, if possible, within the 8 characters of the word, with unused characters left blank.

CONPRO C/HLT, TABLE1; D/HLT, N where location

(CONTROLN) TABLE1 must contain the following:

```
TABLE1    W/00AQTINT
          W/00MOCARO
.         W/000SGPDC     etc.
```

The BCD words given, correspond to the ID's on the control cards required. The ID's must be right adjusted with leading zeros. N = the number of entries in TABLE1. TABLE1 must be followed by a TABLE2, where TABLE2 contains the following:

```
TABLE2    C/JMP, AQTCON; C/JMP, AAA
          C/JMP, MCCON; C/JMP, AAA
          C/JMP, DCCON; C/JMP, AAA   etc.
```

"AQTCON" is a subroutine to convert the control card with the ID "AQTINT"; "MCCON" is a subroutine to convert the control card with the ID "MOCARO", etc. The first entry in TABLE2 must correspond to the first entry in TABLE1. TABLE2 will be N entries long and must be located immediately after TABLE1 (or at location TABLE1 + N).

CARDTBL C/HLT, TABLE1; C/HLT, N where location

(CONTROLN) TABLE1 must contain the following:

```
TABLE1    W/0000000A
          W/0000000B
          W/0000000C   etc.
```

The BCD character given corresponds to the card types in Column 80 of the data cards required by the sequence. The card types must be right adjusted with preceding zeros.

N = the number of entries in TABLE1.

TABLE1 must be immediately followed by TABLE2 (TABLE2 start address = TABLE1 start address + N). TABLE2 contains the following:

```
TABLE2    C/JMP, ACARD; C/JMP, AAA
          C/JMP, BCARD; C/JMP, AAA
          C/JMP, CCARD; C/JMP, AAA   etc.
```

"ACARD" is a subroutine to convert the data card with "A" in Col 80; "BCARD" is a subroutine to convert the data card with "B" in Col 80, etc. The first entry in TABLE2 must indicate a subroutine which converts the data card indicated by the first entry in TABLE1. TABLE2 will be N entries in length.

PROGTBL Program jump table

(CONTROLN)

```
PROGTBL C/JMP, PROG1
        C/JMP, PROG2
             .
             .
             .
        C/JMP, PROGN
        C/JMP, SEAEOJ
```

PROG1 is the starting instruction of the first program in the sequence; "PROG2" is the starting instruction of the second program in the sequence, etc.; and "PROGN" is the starting instruction of the last program in the sequence. This will control the running of programs, as the EXEC will transfer control to the programs sequentially. A jump to SEAEOJ must always be the last entry in the table. Each program above should exit to to "EXEND" or "NXJOB" in the case of an error.

PROGNO (EXEC)
C/HLT, N, C/HLT, where N is the number of the program currently being run. The "Jump and Address" used to transfer control to the program N is at location PROGTBL - 1 + N. PROGNO is incremented by EXEND.

CONBITS (SYS)
SYS master control word, used by SYS for all switches, inhibits, flags, etc. The exact bit assignments are as follows:

bit 0 Always = 0, not used

bit 1-15 Address of next input card to be processed. Used only if input is from magnetic tape. The address is between 600_8 and 770_8. If the address equals 770_8, the next input card is in the next block on Tape 8.

bits 16-20 Not used

bit 21 = 1 if SYS is to search for a "JOB" card, bypassing all other control cards. This search will occur at the next entry to 1NXTCON (M/5).

= 0 if SYS is to retrieve the next sequential control line at the next entry to 1NXTCON.

bit 22 = 1 if input on Tape 8 is in the image mode.

= 0 if input on Tape 8 is in the code mode.

Not used if input is not from magnetic tape.

bit 23 = 1 indicates that the information in the input buffer (M/600 - M/777) has been destroyed, and SYS must reread the last block of data from Tape 8 when the next control line is requested in 1NXTCON.

= 0 if the information in the input buffer has not been destroyed.

Not used if input is not from magnetic tape.

bit 24 Always = 0, not used.

bits 25-39 Left shift required of "PTWORD" to bring the next character to be processed T5 (in bits 0 to 5). If the number = 48, PTWORD has been entirely processed and the next character is in the next word of the control line.

bit 40 Always = 0, not used.

bit 41 = 1 if the accounting clock is in the system (part of the computer complex).

= 0 if no accounting clock is available.

bit 42 = 1 if the next control line is not to be processed immediately upon retrieval. "1NXTCON" will move the next control line into "CONLINE" without processing it, and exit to the "ENTRY JA."

= 0 if the next control line is to be processed immediately after it is retrieved.

bit 43 = 1 if image mode input from magnetic tape should be converted to code mode from Col 1 to Col 24, leaving Col 25 to Col 80 in image mode.

= 0 if image mode input from magnetic tape should be converted to code mode from Col 1 to Col 80. Not used if input is not image mode from magnetic tape.

bit 44 = 1 if a "JOB" card is not legal at this time

(e.g. - during compilation).

= 0 if a "JOB" card is allowed.

bit 45 = 1 if input is from magnetic tape.

bit 46 = 1 if input is from paper tape.

bit 47 = 1 if input is from the flexowriter.

One and only one of the bits 45 to 47 must be set = 1, the other two must be = 0.

SECTION 5

SYSTEM SUBROUTINES

The subrountines available to the Simulation Error Analysis System are classified in four groups: Program,SYS, Philco, and Common. Philco subroutines are accessible from the Philco Subroutine library tape and are read into core by SYS as required. SYS subroutines are in the area of core allocated to SYS; therefore, they are available as long as SYS is retained in core. Program subroutines are those used by one specific program and are compiled with it. In order to conserve core space, the subroutines used by more than one program have been collected into a section called SUBS and are designated the Common System Subroutines.

This section presents the input, output, and calling sequences for all of the common subroutines and a few of the frequently used SYS subroutines. Philco subroutine specifications are a part of the Philco documentation which may be obtained from the Philco Computer Division. The subroutines are presented in alphabetical order and are dated for anticipated revisions and additions.

Input and output quantities will have an associated symbol, when applicable, designating the format of the word. The formats are assigned as follows:

D	- Decimal
F	- Floating Point
0	- Octal
BCD	- Binary Coded Decimal
Tx	- Fixed Point Number with x as the scale factor

Within the calling sequences, many capitalized words are in parentheses to indicate that they are not absolute core locations. These locations must be assigned by the user when calling one of the subroutines.

The subroutines exit to core cells 0 (exponent overflow) or 3 (standard error) bypassing the normal exit when encountering non-recoverable errors. Programs using these routines must preset cells 0 and 3 for error recovery procedure.

PURPOSE:	To read the Accounting Clock and convert it to standard (6-Bit) BCD format.
CALL SEQUENCE:	JMP AKLOK TAM (RESULT 1) TQM (RESULT 2)
INPUTS:	None
OUTPUTS:	RESULT 1 = Δ Δ Δ Δ Y Y M M RESULT 2 = D D H H M M . M
STORAGE REQUIREMENTS:	11 Cells
SUBROUTINES:	Common: KLOK

DESCRIPTION:

Since the Accounting Clock does not contain the year, the last two digits of the year in core location ZYEAR are inserted in bits 24-35 of Result 1, after the Accounting Clock has been converted from 4-bit BCD to 6-bit BCD.

Note: ZYEAR must be updated yearly.

AQTCON
(SEA)
11/30/62

PURPOSE: To convert an AQTINT Control Card.

CALL SEQUENCE:

```
JMP         AQTCON
JAZ         (ERROR)
```

INPUTS:

AQTINT Control Card
CCARD - See Description

OUTPUTS:

OBSDUR	=	Observation duration (minutes)	(F)
TOPT	=	Output option for TBLOC	D(T 47)
OBSOPT	=	Output option for OBSIM	D(T 47)
PERT	=	Perturbations control (for all programs except DC)	D(T 47)
PNCHOBS	=	Punched output option for OBSIM	D(T 47)
CCARD	=	See Description	

STORAGE REQUIREMENTS: 33 Cells

SUBROUTINES: Common: SFXFLTA, SFXINT, CARDER, PANT

DESCRIPTION:

Core location CCARD is checked to see that an AQTINT Control Card has not been converted for the case. If one has been processed, the current one will be rejected with an error comment and the A Reg = 0; if not, the card will be converted, and Bit 1 will be set = 1 in CCARD to indicate a successful conversion. If an error occurs during conversion, an error comment will be printed and the subroutine will exit with the A Reg = 0.

SIMULATION ERROR ANALYSIS SYSTEM **PROGRAM DOCUMENT**
ARCCOS
(SEA)
11/30/62

PURPOSE: To find, the arc cos of a floating point number

CALL SEQUENCE:

```
TMA         (WORD)
JMP         ARCCOS
JMP         (ERROR)
TAM         (RESULT)
```

INPUTS: WORD = Cos (X) (Floating point)

OUTPUTS: RESULT = X, $0 \leq X \leq \pi$ (Floating point)

STORAGE REQUIREMENTS: 14 Cells

SUBROUTINES: Philco: FACOS

DESCRIPTION:

Due to cumulative computational round-off errors, it is possible for the cosine of an angle near 0° to be greater than 1: Therefore a comparison is made between cos (X) and 1.001:

$$\text{if } \cos(X) \quad \text{is} \begin{cases} > 1.001 \rightarrow \text{ERROR} \\ \geq 1.0 \text{ and } \leq 1.001 \rightarrow X = 0 \text{ or } \pi \\ < 1.0 \rightarrow 0 < X < \pi \end{cases}$$

SIMULATION ERROR ANALYSIS SYSTEM **PROGRAM DOCUMENT**

ARCSIN
(SEA)
11/30/62

PURPOSE: To find the arc sin of a floating point number

CALL SEQUENCE:

```
TMA          (WORD)
JMP          ARCSIN
JMP          (ERROR)
TAM          (RESULT)
```

INPUTS: WORD = Sin (X) (Floating point)

OUTPUTS: Result = X, $\pi/2 \leq X \leq \pi/2$ (Floating point)

STORAGE REQUIREMENTS: 16 Cells

SUBROUTINES: Philco: FASIN

DESCRIPTION:

Due to culmulative computational round-off errors, it is possible for the sine of an angle near 90° to be greater than 1. Therefore a comparison is made between sin (X) and 1.001:

$$\text{if } \left|\sin(X)\right| \text{ is } \begin{cases} > 1.001 \rightarrow \text{ERROR} \\ \geq 1.0 \text{ and } \leq 1.001 \rightarrow X = \pm\pi/2 \\ < 1.0 \rightarrow -\pi/2 \leq X < \pi/2 \end{cases}$$

SIMULATION ERROR ANALYSIS SYSTEM

PROGRAM DOCUMENT
ARCTAN
(SEA)
11/30/62

PURPOSE: To find the arc tan of a floating point number

CALL SEQUENCE:

```
TMA         (WORD 1)
TMQ         (WORD 2)
JMP         ARCTAN
TAM         (RESULT)
```

INPUTS: $\left.\begin{array}{l}\text{WORD 1} = \text{Cos (X)} \\ \text{WORD 2} = \text{Sin (X)}\end{array}\right\}$ Floating point

OUTPUTS: RESULT = X, $0 \leq X \leq 2\pi$ (Floating point)

STORAGE REQUIREMENTS: 17 Cells

SUBROUTINES: Philco: FATAN

DESCRIPTION:

If sin (X) = 0 and cos (X) = 0, then X = 0 arbitrarily with no error indication.

BCDCLK
(SEA)
11/30/62

PURPOSE: To read the Accounting Clock and edit it for printing or typing.

CALL SEQUENCE:

```
JMP     BCDCLK
TAM     (RESULT 1)
TQM     (RESULT 2)
```

INPUTS: None

OUTPUTS:

$$\left.\begin{array}{l}\text{RESULT 1} = \text{YY - MM - DD} \\ \text{RESULT 2} = \Delta\text{HH - MM.M}\end{array}\right\} \quad \text{6-Bit BCD}$$

STORAGE REQUIREMENTS: 29 Cells

SUBROUTINES: Common: AKLOK

DESCRIPTION:

This subroutine uses AKLOK to get the Accounting Clock and year in 6-Bit BCD, putting it in the format for printing and typing.

BCDFD
(SEA)
11/30/62

PURPOSE: To convert BCD date to floating point days and fractions since the beginning of the year

CALL SEQUENCE:

```
TMA         (WORD 1)
TMQ         (WORD 2)
JMP         BCDFD
JAZ         (ERROR)
TAM         (RESULT 1)
TQM         (RESULT 2)
```

INPUTS:

WORD 1 = YYMMDDHH } 6-Bit BCD
WORD 2 = MMSS.SSS }

OUTPUTS:

RESULT 1 = Integral number of days } Floating Point
RESULT 2 = Fractional number of days }

STORAGE REQUIREMENTS 71 Cells

SUBROUTINES: Common: SFXFLTA

DESCRIPTION:

If an error is found, the normal exit will be taken with the A Reg = 0

BCK1CRD
(SEA)
11/30/62

PURPOSE: To set input card retrieval (1NXTCON) to reprocess last control line.

CALL SEQUENCE: JMP BCK1CRD

INPUTS: Left address of CONBITS = Location of current control line.

OUTPUTS: Left address of CONBITS = Location of previous control line.

STORAGE
REQUIREMENTS: 20 cells

SUBROUTINES: Common: PANT
SYS: 1TYPOUT

DESCRIPTION:

If input is from magnetic tape, the cell CONBITS is reset to location of previous card image. If input is from the flexowriter, the comment "IMPROPER CONTROL LINE" is typed. If input is from paper tape, the comment "CONTROL LINE ON PAPER TAPE OUT OF ORDER " is printed.

BLKIN
(SYS)
11/30/62

PURPOSE: To call in the block of the SYS Tape specified and transfer control to it.

CALL SEQUENCE: C/JMP, BLKIN;N/#T47
Control returned to ENTRYJA
ENTRYJA not set by JMP BLKIN

INPUTS: # = The decimal number of the block desired
Example: C/JMP, BLKIN; N/7T47 to retrieve the seventh block on tape 1 (SYS tape)

OUTPUTS: Execution of program in block # given

STORAGE REQUIREMENTS: Part of SYS

SUBROUTINES: SYS subs only

DESCRIPTION:

BLKIN reads the SYS tape into cells M/400 to M/577 searching for the block number specified. When block # is found, BLKIN jumps to M/401, and thereby transfers control to the routine in the block specified. Normally control will be returned to the ENTRYJA, thus back to the entry to SYS just prior to the jump to BLKIN.

CARDER
(SEA)
11/30/62

PURPOSE: To output card error comments.

CALL SEQUENCE:

```
(1)  TMA  (WORD)            (3)  JMP   CARDERB
     JMP  CARDER

(2)  TMA  (WORD1)           (4)  TMA  (WORD2)
     JMP  CARDERA                JMP  CARDERC
```

INPUTS: In all cases, XR3 must contain the location of the erroneous card:

```
WORD   =  ? ? ? ? ? ? NN  (BCD Column Number)
WORD1  =  ? ? ? ? ? ? ?N  (BCD Card Type)
WORD2  =  ? ? ? ? ? ? ?N  (BCD Card Type)
```

OUTPUTS: See Description

STORAGE REQUIREMENTS: 48 Cells

SUBROUTINES: Common: PANT, PAGECON

DESCRIPTION:

The subroutine will insert the given parameter into one of the following comments corresponding to the entrance:

(1) - Error in field ending in Col. NN.
(2) - System expects card with N in Col. 80.
(3) - (No comment)
(4) - Card rejected, too many cards of type N.

The program will then output the comment (or no comment) followed by:

```
1   8   16   24   32   40   48   56   64   72   80

(80 Column Card Image ————————————————→ )

1 Line space
```

CARDER subroutines set the no-run switch for the EXEC (bit 0 of EXSWTCH), except for the CARDERC routine.

CARRET
(SEA)
11/30/62

PURPOSE:	To issue a carriage return to the flexowriter.
CALL SEQUENCE:	JMP CARRET
INPUTS:	None
OUTPUTS:	A carriage return character transmitted to the flexowriter
STORAGE REQUIREMENTS:	2 cells
SUBROUTINES:	None

COBGET
(SEA)
11/30/62

PURPOSE: To retrieve one contaminated observation from COBLOC

CALL SEQUENCE:

```
(1)  To initialize:        (2)  Normal:
       JMP  INITCOB                 JMP  COBGET
                             +1H    JMP  (END)
                             +2H         (NORMAL)
```

INPUTS:

COBLOC
LOCCOB = C/HLT, COBLOC; C/HLT, 0
Termination indicator for COBLOC must be 48/0

OUTPUTS:

END = End of contaminated observations
NORMAL - A REG = STAID

```
STAID )
OTYPE |
RANGE |     (See COBLOC format for contents of core
ALPHA >      locations)
DELTA |
RODOT |
T     )
```

STORAGE REQUIREMENTS: 8 Cells

SUBROUTINES: None

DESCRIPTION:

It is necessary to "JMP INITCOB" only before the first "JMP COBGET." INITCOB sets the beginning address of COBLOC; with every call of COBGET, this address is modified to retrieve the next successive contaminated observation. The maximum number of contaminated observations is 500.

COBPCK
(SEA)
11/30/62

PURPOSE: To pack one contaminated observation into COBLOC

CALL SEQUENCE: (1) To initialize: JMP ICOBPCK (2) Normal: JMP COBPCK

INPUTS: LCOBLOC = C/HLT, COBLOC; C/HLT, COBLOC + 3000

STAID
OTYPE
RANGE
ALPHA
DELTA
RODOT
T
} (See COBLOC format for contents of core locations)

OUTPUTS: One contaminated observation in COBLOC (See COBLOC format)

STORAGE REQUIREMENTS: 46 Cells

SUBROUTINES:
Common: PANT, GLOP, PAGECON
Program: OBPCKA, OBPCKS (See OBPCK)

DESCRIPTION:

It is necessary to "JMP ICOBPCK" only before the first "JMP COBPCK". ICOBPCK sets the beginning address of COBLOC. With every call of COBPCK this address is modified in order to pack the next observation successively. 500 observations may be stored in COBLOC; observations in excess of 500 will be printed.

PURPOSE: To convert a SGPDC Control Card.

CALL SEQUENCE:

JMP	DCCON
JAZ	(ERROR)

INPUTS: SGPDC Control Card
CCARD

OUTPUTS:

DCCCP1	- Correction flag override (0,bits 41 through 47 only)
DCCCP2	- Repeat counter override (BCD)
DCCCP3	- Perturbations Control (DC) (D/T47)
DCCCP4	- Output Options (0, bits 42 through 47 only)
DCCCP5	- RMS multiplier override (BCD)
DCCCP6	- First absolute maximum override (F)
DCCCP7	- Second absolute maximum override (F)
DCCCP8	- ABSMX2 override (F)
DCCCP9	- Convergence criterion override (F)
DCCCP10	- DC distribution override (D/T47)

DCS1	- Lower limit (distribution) (F)
DCS2	- Upper limit (distribution) (F)
DCNS	- No. of quantities to compute range of distribution (D/T47)
DCI	- No. of cells in distribution (D/T47)
CCARD	- See Description

STORAGE REQUIREMENTS: 61 Cells

SUBROUTINES: Common: SXSRCH, CARDER, PANT

DESCRIPTION:

Core location CCARD is checked to see that a DC Control Card has not been converted for the case. If one has been processed, the current one will be rejected with an error comment and the A Reg = 0; if not, the card will be converted. If S1 and/or S2 = $\triangle$, then S1 and/or S2 will be set = 3. The subroutine will then make the following tests:

1. S1 and S2 = 0
2. S1 or S2 Neg
3. I Neg
4. I $\geq$ 101

If any of the above are true, or if there was an error on conversion, the card will be rejected with an error comment and the A Reg = 0. If the card conversion is successful, Bit 3 will be set = 1 in core location CCARD.

DELGET
(SEA)
11/30/62

PURPOSE: To retrieve a delta element set from DEBLOC

CALL SEQUENCE:

```
(1)  To initialize        (2)  Normal:
       JMP INITDEL                JMP   DELGET
                            +1H   JMP   (END)
                            +2H   JMP   (NORMAL)
```

INPUTS: LOCDEL = C/HLT, DEBLOC; C/HLT, 0

Termination indicator for DEBLOC must be 48/0
DEBLOC

OUTPUTS: END = End of DEBLOC

```
DELA    }
DELE    }
DELI    }
DELNOD  }  (See DEBLOC format for contents of core location)
DELOM   }
DELMO   }
DELCO   }
```

STORAGE REQUIREMENTS: 17 Cells

SUBROUTINES: None

DESCRIPTION:

It is necessary to "JMP INITDEL" only before the first "JMP DELGET". INITDEL sets the beginning address of DEBLOC. With every call of DELGET this address is modified to retrieve the next successive set of delta elements. The maximum number of sets of delta elements is 100.

DELPCK
(SEA)
11/30/62

PURPOSE: To pack one delta element set into DEBLOC

CALL SEQUENCE: (1) To initialize (2) Normal
JMP IDELPCK JMP DELPCK

INPUTS: LDEBLOC = C/HLT, DEBLOC; C/HLT, DEBLOC + 700

DELA
DELE
DELI
DELNOD
DELOM
DELMO
DELCO
} (See DEBLOC format for contents of core locations)

OUTPUTS: One set of delta elements in DEBLOC (See DEBLOC format)

STORAGE
REQUIREMENTS: 41 Cells

SUBROUTINES: Common: PANT, GLOP, PAGECON

DESCRIPTION:

It is necessary to "JMP IDELPCK" only before the first "JMP DELPCK", IDELPCK sets the beginning address of DEBLOC. With every call of "DELPCK" this address is modified in order to pack the next delta element set successively. 100 sets of delta elements may be stored in DEBLOC. The delta element sets in excess of 100 will be printed.

ECARD
(SEA)
11/30/62

PURPOSE: To control the conversion of nominal and variant element cards 1 and 2 into EBLOC and VEBLOC format.

CALL SEQUENCE: (1) To initialize
JMP ECARDI
(2) Normal
JMP ECARD

INPUTS: XR3 = Location of card image

For E Card processing, the following locations must be set initially:

LEBLOC = C/HLT, EBLOC; C/HLT, EBLOC + 12

SATSTB = C/HLT, EBLOC; C/HLT, 0

For V Card processing, the following locations must be set initially:

LVEBLOC = C/HLT, VEBLOC; C/HLT, VEBLOC + 700

ELMSTB = C/HLT, VEBLOC; C/HLT, 0

LDEBLOC = C/HLT, DEBLOC; C/HLT, DEBLOC + 700

DELSTB = C/HLT, DEBLOC; C/HLT, 0

OUTPUTS:

See Output of VECARD or ELMRED

For E Card processing:

Increments left address of SATSTB by 12
Increments ECOUNT by 1
Sets right address of LOCSAT equal to left address of SATSTB
Sets Bit 1 = 1 in core location CARDS

For V Card processing:

Increments left address of ELMSTB and DELSTB by 7
Increments VECOUNT and DECOUNT by 1
Sets right address of LOCELM and LOCDEL equal to the left address of ELMSTB and DELSTB respectively
Sets Bit 5 = 1 in core location CARDS

STORAGE
REQUIREMENTS: 184 Cells

SUBROUTINES: Common: CARDER, PANT, PAGECON, ELMRED, VECARD

DESCRIPTION:

Nominal element cards must be input before variant element cards.

It is necessary to "JMP ECARDI" only before the first "JMP ECARD" to initially clear buffers. A pair of element cards (cards 1 and 2) must be entered together; if not, the subroutine will search for a pair, while saving the most recent cards in a buffer and printing error comments on those rejected. Having found a pair of element cards, a comparison is made as to whether both cards have the same element type (A, B, C, or D), satellite number, and card type (E or V). If all 3 do not agree, the search for a pair is begun again, while the most recent cards are saved and an error comment is printed. If all 3 agree, card type is tested to determine which routine should unpack the cards: E = ELMRED (nominal element cards) or V = VECARD (variant element cards). If the cards converted were variant element cards, the subroutine will fill DEBLOC with nominal classical elements to be used later to create a true delta element block. Having successfully converted the cards, the subroutine will set counters, strobes and CARDS, then exit. If unsuccessful, the program will take the normal exit, but the appropriate card bit will not be set in core location CARDS.

ELEM1
(SEA)
11/30/62

PURPOSE: To unpack Element Card 1.

CALL SEQUENCE:

```
TMA         (WORD)
JMP         ELEM1
```

INPUTS: WORD = C/HLT, (Elem. Card 1 address); C/HLT, (Output buffer address)
Element Card 1

OUTPUTS: Output buffer-formats for element types A, B, C, D, respectively. In the first word, N = satellite number. All numbers are floating point except those designated as BCD.

TYPE A:	TYPE B:	TYPE C:	TYPE D:
+0ΔΔΔΔΔNNN(BCD)	ΔΔΔΔΔ NNN (BCD)	ΔΔΔΔΔNNN (BCD)	ΔΔΔΔΔNNN(BCD
+1 a(E.R.)	x(KM)	Hq (KM)	a(E R.)
+2 e	y(KM)	e	e
+3 i (Deg)	z(KM)	i (Deg)	i (Deg)
+4 Ω (Deg)	$\dot{x}$(M/sec)	h_o(Deg)	Ω (Deg)
+5 ω (Deg)	$\dot{y}$(M/sec)	U_o(Deg)	ω (Deg)
+6 M_o(Deg)	$\dot{z}$(M/sec)	ZZZZZ PP	h_o(Deg)

$$Z = \begin{matrix}\text{NORTH}\\\text{SOUTH}\end{matrix} \quad P = \left.\begin{matrix}\text{NS}\\\text{SN}\end{matrix}\right\} \text{(BCD)}$$

STORAGE REQUIREMENTS: 24 Cells

SUBROUTINES: Common: SFXFLT

DESCRIPTION:

The subroutine unpacks element card 1 and stores the input numbers into the output buffer as shown above. If an error is found in converting the numbers to floating point, the normal exit will not be taken. Rather, the subroutine will transfer to core location ELERR1, where error recovery must be set up.

PURPOSE: To unpack Element Card 2.

CALL SEQUENCE:

```
TMA      (WORD)
JMP      ELEM2
```

INPUTS: WORD = C/HLT, (Elem. Card 2 address); C/HLT (Output buffer address)
Element Card 2

OUTPUTS: The output buffer formats for the various element types are as follows (the BCD character T is filled by the element type A, B, C, or D, AAA is the element number, NNN is the satellite number, and SSS is the sensor number):

Types A or B:

+0 ETAAANNN (BCD)
+1
+2
+3 (To be filled)
+4
+5
+6
+7 C_o (Days/rev) (F)
+8 t^o (minutes)
+9 (to be filled)
+10 YYMMDDHH (BCD)
+11 MMSS.SSS
REFYEAR 0000000Y
RHOO 00000000
ELMSEN 00000△△△

Types C or D:

+0 ETAAANNN (BCD
+1
+2
+3 (To be filled)
+4
+5
+6
+7 C_o (Days/rev) (F)
+8 To be filled
+9
+10 △△△△△△△△ (BCD)
+11 △△△△△△△△
RHOO 00000000 or ρ_o(km) (F)
ELMSEN 00000SSS (BCD)

STORAGE REQUIREMENTS: 37 Cells

SUBROUTINES: Common: BCDFD, SFXFLT

DESCRIPTION:

The subroutine unpacks element card 2 into the output buffer and core locations as shown above. If errors occur, the normal exit is not taken; rather, the exits indicated will be taken. It is necessary to set up error recovery at these core locations

Invalid epoch time (field 1) ⟶ ELERR2
Invalid C_o (field 2) ⟶ ELERR2A
Invalid ρ_o^o (field 4) ⟶ ELERR2B

PURPOSE: To retrieve one variant element set from VEBLOC

CALL SEQUENCE:

```
(1)  To initialize:        (2)  Normal:
       JMP   INITELM                JMP   ELMGET
                              +1H   JMP   (END)
                              +2H         (NORMAL)
```

INPUTS: VEBLOC
LOCELM = C/HLT, VEBLOC; C/HLT, 0
Termination indicator for VEBLOC must be 48/0

OUTPUTS: END = End of VEBLOC

AXNO
AYNO
HXO
HYO
HZO
XLO
CO

(See VEBLOC format for contents of core locations)

STORAGE REQUIREMENTS: 8 Cells

SUBROUTINES: Program: NXSAT (See EUNPCK)

DESCRIPTION:

It is necessary to "JMP INITELM" only before the first "JMP ELMGET". INITELM sets the beginning address of VEBLOC. With every call of ELMGET, this address is modified to retrieve a successive variant element set. The maximum number of sets of variant elements is 100.

ELMOUT
(SEA)
11/30/62

PURPOSE:	To print elements - Type A, B, C, or D
CALL SEQUENCE:	JMP ELMOUT
INPUTS:	ELMTYP = 0000000N, N = A, B, C, or D (BCD) EBUF (Output Buffer)
OUTPUTS:	Contents of EBUF
STORAGE REQUIREMENTS:	149 Cells
SUBROUTINES:	Common: GLOP, PAGECON, PANT Program: ELOUTA, ELOUTB, ELOUTC, ELOUTD

DESCRIPTION:

This subroutine tests ELMLYP to determine which of the following program subroutines to use to output the values in EBUF:

Element Type	A	-	ELOUTA
Element Type	B	-	ELOUTB
Element Type	C	-	ELOUTC
Element Type	D	-	ELOUTD

(Description continued)

	Element Type		Element Type		Element Type		Element Type	
	A		B		C		D	
EBUF	a	(E.R.).	x	(Km)	a	(E.R.)	a	(E.R.)
+ 1	e		y	(Km)	e		e	
+ 2	i	(Deg)	z	(Km)	i	(Deg)	i	(Deg)
+ 3	Ω	(Deg)	$\dot{x}$	(m/sec)	Ω	(Deg)	Ω	(Deg)
+ 4	ω	(Deg)	$\dot{y}$	(m/sec)	ω	(Deg)	ω	(Deg)
+ 5	M_o	(Deg)	$\dot{z}$	(m/sec)	M_o	(Deg)	M_o	(Deg)
+ 6	C_o	(Days/rev)	C_o	(Days/rev)	C_o	(Days/rev)	C_o	(Days/rev)
+ 7	YYMMDDHH		YYMMDDHH		△———△		△———△	
+ 8	MMSS.SSS		MMSS.SSS		△———△		△———△	
+ 9					STAID		STAID	
+ 10					ϕ	(Deg)	ϕ	(Deg)
+ 11					λ	(Deg)	λ	(Deg)
+ 12					H	(Km)	H	(Km)
+ 13					Hq	(Km)	a	(E.R.)
± 14					e		e	
+ 15					i	(Deg)	i	(Deg)
+ 16					h_o	(Deg)	Ω	(Deg)
+ 17					v_o	(Deg)	ω	(Deg)
+ 18					(NORTH△△△ SOUTH△△△)		h_o	(Deg)
+ 19					(NS△△△△△△ SN△△△△△△)		ρ_o	(Km)

PURPOSE: To pack one variant element set into VEBLOC

CALL SEQUENCE: (1) To initialize (2) Normal:

```
(1)  To initialize        (2)  Normal:
       JMP   IELMPCK             JMP   ELMPCK
```

INPUTS: LVEBLOC = C/HLT, VEBLOC; C/HLT, VEBLOC + 700

AXNO
AYNO
HXO
HYO
HZO
XLO
CO

(See VEBLOC format for contents of core locations)

OUTPUTS: One set of variant elements in VEBLOC (See VEBLOC format)

STORAGE REQUIREMENTS: 39 Cells

SUBROUTINES: Common: GLOP, PANT, PAGECON

DESCRIPTION:

It is necessary to "JMP IELMPCK" only before the first "JMP ELMPCK". IELMPCK sets the beginning address of VEBLOC; with every call of ELMPCK, this address is modified in order to pack the next variant element set successively. 100 sets of variant elements (700 words) may be stored in VEBLOC. The variant element sets in excess of 100 will be printed.

PURPOSE: To control the conversion of a pair of element cards into EBLOC format.

CALL SEQUENCE:

```
TMA        (WORD)
JMP         ELMRED
JAZ        (ERROR)
```

INPUTS: WORD = C/HLT, (Address of element card); C/HLT, (Block address)

OUTPUTS: $\underline{N}$, $\underline{M}$ elements, θ_{t_o}, t_o, epoch, element type, in EBLOC

STORAGE REQUIREMENTS: 209 Cells

SUBROUTINES: Common: ELEM1, ELEM2, ETYPA, ETYPB, ETYPC, ETYPD, SFXFLT, SENGET, STLC, SEPSUB, PANT, CARDER, PAGECON, BCDFD, ARCTAN, ARCSIN, ARCCOS

Philco: FSQRT, FSIN, FCOS

DESCRIPTION:

The subroutine will validate the element type (A, B, C, D); if it is invalid, the normal exit will be taken with the A Reg = 0 after an error comment has been printed. If it is valid, the element cards will be unpacked and converted to $\underline{N}$ $\underline{M}$ elements. These will be stored in EBLOC, as will the other values given in the EBLOC format. The subroutine will then set up the output buffer (EBUF) so that the one of following may be printed by another subroutine ELMOUT:

1. Element type A - classical elements
2. Element type B - $\underline{r}$, $\dot{\underline{r}}$ elements
3. Element type C - orbit-station elements, classical elements, sensor coordinates
4. Element type D - radar-oriented elements, classical elements, sensor coordinates

ERPCON
(SEA)
11/30/62

PURPOSE: To convert an Error Prediction (ERPRED) Control Card.

CALL SEQUENCE:

```
JMP        ERPCON
JAZ        (ERROR)
```

INPUTS: Error Prediction (ERPRED) Control Card
CCARD - See Description

OUTPUTS:

CASETYP	ERPDOPT	See Dictionary
OTPFLAG	ERPS1	
APMFLAG	ERPS2	
DPVFLAG	ERPNS	
CAPPI	ERPI	
DELCAPP	OPOUT1	
CAPPF	OPOUT2	

STORAGE REQUIREMENTS: 72 Cells

SUBROUTINES: Common: SXSRCH, CARDER, PANT

DESCRIPTION:

Core location CCARD is checked to see that an ERPRED Control Card has not been converted for the case. If one has been processed, the current one will be rejected with an error comment and the A Reg = 0; if not, the card will be converted. If S1 and/or S2 = Δ, then S1 and/or S2 will be set = 3. The subroutine will then make the following tests:

1. S1 and S2 = 0
2. S1 or S2 NEG
3. I NEG
4. $I \geq 101$

If any of the above are true, or if there was an error on conversion, the card will be rejected with an error comment and the A Reg = 0. If the card conversion is successful, Bit 5 will be set = 1 in core location CCARD.

PURPOSE: To convert classical elements to N M elements.

CALL SEQUENCE:

```
TMA     (WORD)
JMP     ETYPA
```

INPUTS: WORD = C/HLT, (Input Buffer Address); C/HLT,(Output Buffer Address)

Input Buffer (all floating point):

+ 0	Not used
+ 1	a (Earth radii)
+ 2	e
+ 3	i (Deg)
+ 4	Ω (Deg)
+ 5	ω (Deg)
+ 6	M_o (Deg)

OUTPUTS: Output Buffer (all floating point):

+ 0	Not used	
+ 1	$a_{x_{N_o}}$	
+ 2	$a_{y_{N_o}}$	
+ 3	h_{x_o}	$(\text{Earth radii})^{1/2}$
+ 4	h_{y_o}	$(\quad " \quad)^{1/2}$
+ 5	h_{z_o}	$(\quad " \quad)^{1/2}$
+ 6	L_o	(Radians)

STORAGE REQUIREMENTS: 41 Cells

SUBROUTINES: Philco: FSIN, FCOS, FSQRT

DESCRIPTION:

Three input values are validated. If an error is found, the normal subroutine exit is not taken, but rather the exits indicated. It is necessary to set up error recovery at these core locations.

If $a < 1.0$, then ⟶ ELERRA
If $e \geq 1.0$, then ⟶ ELERRE
If $i < 0$ or $i > \pi$, then ⟶ ELERRI

Having successfully validated these quantities, the subroutine will convert classical elements to N M elements with the formulas presented in the initialization portion of Section 3.

PURPOSE: To convert $\underline{r}$, $\underline{\dot{r}}$ elements to $\underline{N}$ $\underline{M}$ elements

CALL SEQUENCE:

```
TMA     (WORD)
JMP     ETYPB
```

INPUTS: WORD = C/HLT, (Input Buffer Address); C/HLT, (Output Buffer Address)

Input Buffer (all floating point):

+ 0 Not used
+ 1 x (Earth radii)
+ 2 y (Earth radii)
+ 3 z (Earth radii)
+ 4 $\dot{x}$ (Radii/ke^{-1} min)
+ 5 $\dot{y}$ (Radii/ke^{-1} min)
+ 6 $\dot{z}$ (Radii/ke^{-1} min)

OUTPUTS:

Output Buffer (all floating point):

+ 0 Not used
+ 1 a_{xno}
+ 2 a_{yno}
+ 3 h_{xo} (Earth radii)$^{1/2}$
+ 4 h_{yo} (Earth radii)$^{1/2}$
+ 5 h_{zo} (Earth radii)$^{1/2}$
+ 6 L_o (Radians)

STORAGE REQUIREMENTS: 100 Cells

SUBROUTINES: Common: ARCTAN
Philco: FSQRT

DESCRIPTION:

The subroutine coverts $\underline{r}$, $\underline{\dot{r}}$ elements to $\underline{N}$ $\underline{M}$ elements with the formulas given in Section 3.

PURPOSE: To convert orbit-station elements to classical elements

CALL SEQUENCE:

TMA	(WORD)
JMP	ETYPC

INPUTS: WORD = C/HLT, (Input Buffer Address); C/HLT, (Output Buffer Address)

Input Buffer:

+ 0	Not used	
+ 1	H_q (Km)	(F)
+ 2	e	
+ 3	i (Degrees)	
+ 4	h_o (Degrees)	
+ 5	v_o (Degrees)	
+ 6	NORTH Δ NS SOUTH Δ SN	(BCD)

OUTPUTS: Output Buffer (all floating point):

+ 0	Not used	
+ 1	a	(Earth radii)
+ 2	e	
+ 3	i	(Degrees)
+ 4	Ω	(Degrees)
+ 5	ω	(Degrees)
+ 6	M_o	(Degrees)

STORAGE REQUIREMENTS: 80 Cells

SUBROUTINES:

Common:	ARCSIN, ARCTAN
Philco:	FSIN, FCOS, FSQRT

DESCRIPTION:

Two input quantities are validated.If an error is found, the normal subroutine exit is not taken, rather the exits indicated will be taken. It is necessary to set up error recover at these core locations.

If $i = 0$, then ELERRI

If $e \geq 1$, then ELERRE

Having successfully validated these quantities, the subroutine will convert orbit-station elements to classical elements with the formulas given in Section 3. During the conversion, it is possible to find an error in computing v_o , β_o, or K_o. Therefore it is necessary to set up error recovery at core locations UOERR, BOERR, KOERR. These exits will be taken if the sine of the respective angle is > 1. Epoch time is zero and is stored in appropriate core location.

SIMULATION ERROR ANALYSIS SYSTEM — PROGRAM DOCUMENT

ETYPD
(SEA)
11/30/62

PURPOSE: To convert radar oriented elements to classical elements.

CALL SEQUENCE:

```
TMA       (WORD)
JMP       ETYPD
```

INPUTS:

WORD = C/HLT, (Input Buffer Address)
       C/HLT, (Output Buffer Address)

Input Buffer (all floating point):

+ 0	Not used
+ 1	a (Earth radii)
+ 2	e
+ 3	i (Degrees)
+ 4	Ω (Degrees)
+ 5	ω (Degrees)
+ 6	h_o (Degrees)
RHOO	ρ_o (Km) (F)

OUTPUTS:

Output Buffer (all floating point):

+ 0	Not used
+ 1	a (Earth radii)
+ 2	e
+ 3	i (Degrees)
+ 4	Ω (Degrees)
+ 5	ω (Degrees)
+ 6	M_o (Degrees)

STORAGE: 140 Cells

SUBROUTINES:

Common: ARCCOS, ARCTAN
Philco: FSIN, FCOS, FSQRT, FATAN

DESCRIPTION:

Three input quantities are validated. If an error is found, the normal subroutine exit is not taken, rather the exits indicated will be taken. It is necessary to set up error recovery at these core locations

If $a < 1$, then ELERRA
If $e \geq 1$, then ELERRE
If $i = 0$, then ELERRI

(Description continued)

Having successfully validated these quantities, the subroutine will convert radar oriented elements to classical elements with the formulas given in Section 3. During the conversion, it is possible to encounter $\rho_{min} > \rho_o > \rho_{max}$, $|\cos v_o| > 1$, $|\cos \epsilon| > 1$, $|\sin \delta_g| > 1$, or $K_1 < 0$. Therefore it is necessary to set up the respective error recovery locations ELERRO, VOERR, ELERREP, ELERRDG, and ELERRK.

EUNPCK
(SEA)
11/30/62

PURPOSE: To retrieve and unpack one element set from EBLOC

CALL SEQUENCE:

```
            JMP   EUNPCK
      +1H   JMP   (ERROR)
      +2H         (NORMAL)
```

INPUTS: EBLOC
LOCSAT = C/HLT, EBLOC; C/HLT, 0

OUTPUTS:

```
SATN      HXO      CO
ELNO      HYO      TO         (See EBLOC format for
AXNO      HZO      THGRTO      contents of core locations)
AYNO      XLO      YY
                   MM
```

STORAGE REQUIREMENTS: 36 Cells

SUBROUTINES: Program: INITSAT, NXSAT

DESCRIPTION:

This subroutine uses "INITSAT" to initialize retrieval and "NXSAT" to retrieve the element set and other values from EBLOC. The ERROR return is obtained if the first word of EBLOC is zero, indicating no elements.

FKLOK
(SEA)
11/30/62

PURPOSE: To read the accounting clock and convert it to floating point days

CALL SEQUENCE:

```
JMP     FKLOK
TAM     (RESULT 1)
TQM     (RESULT 2)
```

INPUTS: None

OUTPUTS:

```
RESULT 1 = Accounting clock - (floating point)
RESULT 2 = Accounting clock (4-bit BCD) - OYYMMDDHHMM ^ M
                                                      ↑
                                               Implied Decimal
```

STORAGE REQUIREMENTS: 11 Cells

SUBROUTINES: Common: AKLOK, BCDFD

DESCRIPTION:

The accounting clock reading is converted to floating point days since the beginning of the year and left in the A register; the accounting clock is left in the Q register in 4-bit BCD.

FLOAT
(SEA)
11/30/62

PURPOSE:	Process a FLOAT card.
CALL SEQUENCE:	JMP FLOAT
INPUTS:	XR3 = card location
OUTPUTS:	Floating correction
STORAGE REQUIREMENTS:	10 Cells
SUBROUTINES:	Common: SFXFLT SYS: STRIP

DESCRIPTION:

Uses the address in Col. 9-16, converted to octal, as the location to be corrected. Columns 25 to 40 are converted to floating point binary using SFXFLT and stored in the specified location.

GETBL
(SYS)
11/30/62

PURPOSE: To set mode of subroutine SCANON to ignore or retrieve blanks

CALL SEQUENCE:

(1) To retrieve blanks:

JMP GETBL

(2) To ignore blanks:

JMP IGBL

INPUTS: None

OUTPUTS: None

STORAGE REQUIREMENTS: Part of SYS

SUBROUTINES: None

PURPOSE: To convert specified data words to BCD suitable for output purposes and output the converted group using PANT.

CALL SEQUENCE:

```
(1)  TMA      CWORD
     JMP.     GLOP.GLOP
```

This prepares specified data words for printing and jumps to PANT to put the line on magnetic tape.

```
(2)  TMA      CWORD
     JMP      GLOP.TGLOP
```

This prepares specified data words for TTY punching and jumps to TPANT to put image on magnetic tape.

```
(3)  TMA      CWORD
     JMP      GLOP.PGLOP
```

This prepares specified data words for card punching and jumps to PPANT to put image on magnetic tape.

INPUTS:

GLOP uses an indirect calling sequence, i.e., the parameters need not be in line in the user's coding. The control word is of the form:

```
L CWORD    C/HLT,0;C/TIJL,FWORD, Xi
```

FWORD is the first of a string of format words located in sequential core locations, and Xi may be any non-counting index register, except 1,2,3,4. If the programmer desires to put more than one sequence of format words together to specify a line of output, the code word has the format:

```
L CWORD    C/TMA,CWORD2;C/TIJL,FWORD2,Xi
```

CWORD2 is the location of another code word of the same form. When the left half of CWORD is zero, the end of the string of code words is signified. A string of CWORDs need not be in sequential core locations.

Example

```
                TMA        CWORD
                JMP        GLOP.GLOP

     CWORD      C/TMA,CWORD2;C/TIJL,FWORD

     CWORD2     C/TMA,CWORD3;C/TIJL,FWORD2

     CWORD3     C/HLT,0;C/TIJL,FWORD3
```

This calling sequence would cause GLOP to use as format words the three strings beginning at locations FWORD,FWORD2,FWORD3. The strings of format words would be processed in the order given. See Description for the details of the format words.

OUTPUTS: A 15 word output buffer (GLOP.BUFF) moved to the proper buffer in PANT.

STORAGE REQUIREMENTS: 210 Cells

SUBROUTINES: Common: PANT

DESCRIPTION:

The format words specify the various types of output desired and how they are to be positioned on the line. In general, the format words are of the form:

```
L FWORD    MODE    P*128 + F
           FLAG    DATA, Xi
```

Location FWORD is the first word of a sequence of format words - one word for each field on the line. The parameters which must be specified by the format words are:

1. P - the position of the field on the line.
2. F - the number of characters in the field.

3. DATA,Xi - the location of the data.
4. MODE - the mode (0 to 6) of conversion to alphanumeric.
5. FLAG - the end of the format sequence.
6. The scale factor, if a fixed point number is involved.

Each conversion mode requires its own parameters; however, some parameters are common to all modes and will be discussed here.

The parameter P is used in all modes and is defined as the right-most print position of the data being processed. Its range is $0<P\leq120$ for hard copy, $0<P\leq80$ for cards, and $0<P\leq72$ for teletype. The burden is on the user to insure enough room on the line for the information, without overlap. If an attempt is made to put more than one character in any one print position, the last character specified will be the one used.

The end of the format sequence is indicated by the command in the right half of the format word, "FLAG". If this command is "TMA" the GLOP conversion continues with the format word in the next sequential core location. If it is "CAM" the sequence is terminated and GLOP processes the next control word. This is specified in the left half of the previous control word. If the left half of the previous control word was a "HLT,0," all conversions have been finished and the line will be transferred from the line buffer to the output buffer. If the left half of the last control word was a "TMA, CWORDn", CWORDn is processed by GLOP.

Each conversion mode will now be taken up in turn.

Mode 0 - Alphanumeric Data - No conversion

```
Format:  HLT      P*128 + F
         FLAG     DATA, Xj
```

F is the number of characters to be placed into the line buffer starting with the first character in location DATA, X_j. The last of these characters will be in position P. X_j may be any non-counting index register except 1X, 2X, 3X or 4X.

The range of F<is 0 F≤120.

<u>Mode 1</u> - Fixed point binary to fixed point decimal.

```
Format:   JBT       P * 128 + F
          FLAG      DATA, Xj
          D/X
```

F is the number of fractional digits to the right of the decimal point, the last being in print position P. As many places as are necessary to print the number, and its sign, are taken to the left of the decimal point. The total number of digits, both integral and fractional may not exceed 14.

X is the scale factor of the fixed point binary number in location DATA,X_j. X_j may be any non-counting index register except 1X, 2X, 3X or 4X.

<u>Mode 2</u> - Floating point binary to fixed point decimal.

```
Format:   ICOZ      P * 128 + F
          FLAG      DATA, Xj
```

The format for Mode 2 is exactly the same as for Mode 1 except for the scale factor, which is not used. If F is zero, no decimal point is printed.

<u>Mode 3</u> - Fixed point binary to floating point decimal.

```
Format:   NOPL      P * 128 + F
          FLAG      DATA, Xj
          D/X
```

F is the total field size, with the right-most character in position P. The printed number will look like ±mmm...mmm±ee, where m is a mantissa digit and e is an exponent digit. The field is from the leading sign to the final e, and may not exceed 16.

Only the number of exponent digits required to expressed the number will be printed. If the exponent is zero, the mantissa takes the entire field.

X is the scale factor of the fixed point binary number in location DATA, X_j. X_j may be any non-counting index register except 1X, 2X, 3X or 4X.

Mode 4 - Floating binary to floating point decimal.

```
Format:   TIO      P * 128 + F
          FLAG     DATA, Xj
```

The format for Mode 4 is exactly the same as for Mode 3, with the exception of the scale factor, which is not used.

Mode 5 - Fixed binary integer to decimal integer.

```
Format:   TCM      P * 128 + F
          FLAG     DATA, Xj
```

F is the number of right shifts required to scale the word in location DATA, X_j 47; i.e., if one has an integer in the right half address (scaled 39), an F of 8 will cause the number to be scaled 47 before conversion. If the word in DATA, X_j is scaled 47, F should be = 0. The integer will be printed as a decimal integer with the low order digit in position P. As many characters as necessary will be used to express the number. Leading zeros or a plus sign will not be printed; if the number is negative, a leading minus sign will be printed.

Mode 6 - Binary to octal.

```
Format:   SKC      P * 128 + F
          FLAG     DATA, Xj
```

The binary word will be printed in octal in the form xxxx xxxx xxxx xxxx. Nineteen print positions are used. The right-most octal digit is placed in position P . F is not used.

HEAD
(SEA)
11/30/62

PURPOSE: To output page headings on printed output.

CALL SEQUENCE: JMP HEAD

INPUTS: None

OUTPUTS: Page eject, system page heading, 2 line spaces

STORAGE
REQUIREMENTS: 54 Cells

SUBROUTINES: Common: PANT, GLOP

PURPOSE: To control the conversion of a simulation parameter card into IBLOC format.

CALL SEQUENCE: JMP ICARD

INPUTS: XR3 = Location of card image
LIBLOC = C/HLT, IBLOC; C/HLT, IBLOC + 700
ISTB = C/HLT, IBLOC; C/HLT, 0 (initially)

OUTPUTS: Left address of ISTB is incremented by 7 and the right address of LOCI is set equal to it

ICOUNT is incremented by 1

Sets I card bit (Bit 3 = 1) in core location CARDS

STORAGE REQUIREMENTS: 13 Cells

SUBROUTINES: Common: SPARM, CARDER

DESCRIPTION:

The output listed is obtained only if no error is encountered during card conversion. The maximum number of cards that will be accepted is 100; those in excess of 100 will be rejected with an error comment.

Also see SPARM.

IGET
(SEA)
11/30/62

PURPOSE: To retrieve one set of simulation parameters from IBLOC

CALL SEQUENCE:

```
(1)  To initialize:           (2)  Normal:
          JMP        INITI                JMP         IGET
                                   +1H    JMP         (END)
                                   +2H                (NORMAL)
```

INPUTS:

LOCI = C/HLT, IBLOC; C/HLT, 0
Termination indicator for IBLOC must be 48/0
ELMTYP = 0000000X, X = Element type (A, B, C, D)
IBLOC

OUTPUTS:

END = End of IBLOC, NORMAL: A Reg = STAID

STAID
OTYPE
VTYPE
DIRI
DIRF
NUSEG
SEGNOS
HI
HF
IDELTAT
ROMAX

(See IBLOC format for contents of core locations)

STORAGE REQUIREMENTS: 30 Cells

SUBROUTINES: None

DESCRIPTION:

It is necessary to "JMP INITI" only before the first "JMP IGET". INITI sets the beginning address of IBLOC. With every call of IGET this address is modified to retrieve the next set of simulation parameters. If ELMTYP is not available, it is necessary to "JMP EUNPCK" to retrieve it. ELMTYP is tested; if it is = C or D, VTYPE is made non-visual (=0). The maximum number of sets of simulation parameters is 100.

KLOK
(SEA)
11/30/62

PURPOSE: To read the Accounting Clock in 4-bit BCD format.

CALL SEQUENCE:

```
JMP       KLOK
TAM       (RESULT)
```

INPUTS: None

OUTPUTS:

```
Result =  OYYMMDDHHMM M (4-bit BCD)
                     ↑
               Implied decimal
```

STORAGE REQUIREMENTS: 10 Cells

SUBROUTINES: None

DESCRIPTION:

Since the Accounting Clock does not contain the year, the last two digits of the year in core location ZYEAR is converted to 4-bit BCD and inserted in bits 4-11 of the result. ZYEAR must be updated yearly.

PURPOSE: To form a least squares matrix and solve the matrix equation Ax = B for x; where

$$A = \begin{bmatrix} \sum C_1 C_1 & \sum C_1 C_2 \cdots\cdots & \sum C_1 C_n \\ \sum C_2 C_1 & \sum C_2 C_2 \cdots\cdots & \sum C_2 C_n \\ \vdots & \vdots & \vdots \\ \sum C_n C_1 & \sum C_n C_2 & \sum C_n C_n \end{bmatrix}, \quad B = \begin{bmatrix} \sum C_1 C_n + 1 \\ \sum C_2 C_n + 1 \\ \vdots \\ \sum C_n C_n + 1 \end{bmatrix}$$

CALL SEQUENCE:

(1) To set n

```
TMA         CWORD
JMP         CHGNXN
```

CWORD = C/HLTL, n ($n \leq 7$)

(2) To initialize the routine (Zero A and B matrices)

```
JMP         LSQ
```

This will jump to a routine call FUNCT for the first set of terms C_1 to C_{n+1}

(3) To add to the matrix A and B

```
JMP         LSQR
```

C_1 to C_{n+1} are to be located in cells TERMS to TERMS + n. This will jump to FUNCT for the next set of C's.

(4) To solve the matrix equation

```
JMP         LSQS
```

Solves the matrix equation, leaving the solution in LSQX to LSQX + n - 1.

(Continued)

STORAGE
REQUIREMENTS: 141 Cells

SUBROUTINES: Philco: FMAIN, FMAMU

DESCRIPTION:

The top half of matrix A is accumulated at each entry to LSQR. When entry is made to LSQS, the terms of A are placed in the bottom half, then entry is made to the standard Philco matrix inversion and multiplication subroutines.

PURPOSE: To control the conversion of a standard deviation and bias card into SGBLOC format.

CALL SEQUENCE: JMP MCARD

INPUTS: XR3 = Location of card image
LSGBLOC = C/HLT, SGBLOC; C/HLT, SGBLOC + 900
SIGSTB = C/HLT, SGBLOC; C/HLT, 0 (Initially)

OUTPUTS: The left address of SIGSTB is incremented by 9 and the right address of LOCSIG is set equal to it.

SGCOUNT is incremented by 1

Sets M card bit (bit 4 = 1) in core location CARDS

STORAGE REQUIREMENTS: 11 Cells

SUBROUTINES: Common: CARDER, STDEV

DESCRIPTION:

The output listed is obtained only if no error is encountered during card conversion. The maximum number of cards that will be accepted is 100; those in excess of 100 will be rejected with an error comment.

Also see STDEV

MCCON
(SEA)
11/30/62

PURPOSE: To convert a Monte Carlo (MOCARO) Control Card

CALL SEQUENCE:

```
JMP     MCCON
JAZ     (ERROR)
```

INPUTS: MOCARD Control Card
CCARD
KMAX = D/100(T47)

OUTPUTS: K - Number of contaminated observations to be generated (T47)
FIRSTRN - Beginning random number (T47)
COFLAG - Monte Carlo output option (T47)

STORAGE REQUIREMENTS: 35 Cells

SUBROUTINES: Common: SFXINT, CARDER, PANT

DESCRIPTION:

Core location CCARD is checked to see that a Monte Carlo Control Card has not been converted for the case. If one has been processed, the current one will be rejected with an error comment and the A Reg = 0; if not, the card will be converted. K will be tested to see that it is less than KMAX. If it is not, or if there is an error on conversion, an error comment will be printed and the subroutine will exit with A Reg = 0. If the card conversion is successful, Bit 2 will be set = 1 in core location CCARD.

PURPOSE: To compute position and velocity at a specific time.

CALL SEQUENCE: JMP MNREP

INPUTS (All floating point):

XJGRCF	$k_e \sqrt{\mu} J a_e^2$	0 or $1.20717162 \times 10^{-4} \frac{(\text{radii})^{7/2}}{\text{min}}$
CO	Perigee drag parameter	C_o (days/rev)
HXO	Inertial X component of angular momentum	$h_x \frac{(\text{Earth radii})^2}{k_e^{-1} \text{ min}}$
HYO	Inertial Y component of angular momentum	$h_y \frac{(\text{Earth radii})^2}{k_e^{-1} \text{ min}}$
HZO	Inertial Z component of angular momentum	$h_z \frac{(\text{Earth radii})^2}{k_e^{-1} \text{ min}}$
AXNO	$\underline{N}$ component of $\underline{a} = e\underline{P}$	a_{xN}
AYNO	$\underline{M}$ component of $\underline{a} = e\underline{P}$	a_{yN}
XLO	Mean longitude	L(Radians)
T	Elapsed time since epoch conditions	$t - t_o$ (min)

OUTPUTS (All floating point):

X	Inertial X component of position at time t	x Earth radii
Y	Inertial Y component of position at time t	y Earth radii

OUTPUTS (All floating point): (continued)

Z	Inertial Z component of position at time t	z	Earth radii
XDOT	Inertial X component of velocity at time t	$\dot{x}$	$\frac{\text{Earth radii}}{k_e^{-1} \text{ min}}$
YDOT	Inertial Y component of	y	$\frac{\text{Earth radii}}{k_e^{-1} \text{ min}}$
ZDOT	Inertial Z component of	z	$\frac{\text{Earth radii}}{k_e^{-1} \text{ min}}$

STORAGE REQUIREMENTS: 200 Cells

SUBROUTINES: Philco: FSIN, FCOS, FSORT

DESCRIPTION:

Given the six orbital quantities, a_{xN}, a_{yN}, h_x, h_y, h_z and L, along with the quantities $t - t_o$, C_o, and XJGRCF, MNREP will compute and output position $\underline{r}$ and velocity $\dot{\underline{r}}$. The input quantity, XJGRCF = $k_e \sqrt{\mu}\ J a_e^2$, controls the method of computation of $\underline{r}$ and $\dot{\underline{r}}$ (i.e., XJGRCF = $1.20717162 \times 10^{-4}$ includes the perturbation effects in the computation of $\underline{r}$ and $\dot{\underline{r}}$, XJGRCF = 0 bypasses all perturbation effects and computes the two body $\underline{r}$ and $\dot{\underline{r}}$). By means of initializing checks, the routine establishes whether the incoming element set had been used previously. If so, it bypasses a portion of the computation that would be redundant, thus saving computation time.

The formulation used in the MNREP subroutine is presented below. Reference may also be made to the enclosed flow diagram for the logical sequence of computation.

(1) Compute the preliminary quantities:

$$p_o = h_{x_o}^2 + h_{y_o}^2 + h_{z_o}^2$$

$$e_o^2 = a_{x_{N_o}}^2 + a_{y_{N_o}}^2$$

$$a_o = p_o/(1 - e_o^2)$$

$$n_o = k_e\sqrt{\mu} \,/\, a_o^{3/2}$$

$$\underline{W}_o = \underline{h}_o / \sqrt{p_o}$$

$$\sin i = \sqrt{1 - W_{z_o}^2}$$

$$\cos i = W_{z_o}$$

$$i = \tan^{-1}(\sin i/\cos i) \quad o \leq i \leq \pi$$

$$\cos \Omega_o = -W_{y_o}/\sin i$$

$$\sin \Omega_o = W_{x_o}/\sin i$$

$$\Omega_o = \tan^{-1}(\sin \Omega_o/\cos \Omega_o)$$

$$q_o = a_o\,(1 - e_o)$$

(2) For simplified general perturbations computation, i.e. when XJGRCF ≠ 0, compute the perturbative coefficients c", k_e K' and k_e L'_{so}:

$$c'' = -360\, n_o^2\, C_o/\pi^2$$

$$k_e J' = k_e \sqrt{\mu}\, J_{a_e}^2 / p_o^{7/2}$$

$$k_e L'_{so} = [k_e \sqrt{\mu}\, J_{a_e}^2 / p_o^{7/2}]\, [3 - 5e_o^2 - \left|\cos i\right| (1 - \tfrac{3}{2} e_o^2) - \sin^2 i (4 - \tfrac{27}{4} e_o^2)]$$

In two body computation, i.e. when XJGRCF = 0, the above coefficients are identically zero.

(3) Compute the perturbed orbital elements:

$$a = a_o \left[1 - \frac{4}{3}\, c'' \,(t - t_o)\right]$$

$$n = k_e \sqrt{\mu} / a^{3/2}$$

$$e = 1 - a_o/a, \text{ FOR } a \geq q_o$$

$$= 0 \quad , \text{ FOR } a < q_o$$

$$p = q_o\,(1 + e)$$

$$\frac{d\Omega}{dt} = -k_e\, J' \cos i \,(1 - \frac{3}{2}\, e^2)$$

$$\Omega = \Omega_o + \frac{d\Omega}{dt}\,(t - t_o)$$

$$\frac{d\omega}{dt} = k_e \frac{J'}{2} (5 \cos^2 i - 1)(1 - \frac{3}{2} e^2)$$

$$\omega_s = \omega - \omega_o = \frac{d\omega}{dt} (t - t_o)$$

$$a_{xN} = \frac{e}{e_o} (a_{x_{N_o}} \cos \omega_s - a_{y_{N_o}} \sin \omega_s)$$

$$a_{yN} = \frac{e}{e_o} (a_{xN_o} \sin \omega_s + a_{yN_o} \cos \omega_s)$$

$$L = L_o + (n_o + k_e L'_{so}) (t - t_o) + n_o c'' (t - t_o)^2$$

(4) Compute U and the components of the vectors $\underline{M}$, $\underline{N}$, $\underline{W}$:

$$W_x = \sin \Omega \sin i$$

$$W_y = \cos \Omega \sin i$$

$$W_z = \cos i$$

$$M_x = - \sin \Omega \cos i$$

$$M_y = \cos \Omega \cos i$$

$$M_z = \sin i$$

$$N_x = \cos \Omega$$

$$N_y = \sin \Omega$$

$$N_z = 0$$

$$U = L - \Omega, \text{ IF } W_z \geq 0$$

$$= L + \Omega, \text{ IF } W_z < 0$$

(5) Compute $(E + \omega)$ by iteration:

$$E + \omega = U + a_{x_N} \sin (E + \omega) - a_{y_N} \cos (E + \omega)$$

(6) Compute the additional auxiliary quantities:

$$e \cos E = a_{xN} \cos (E + \omega) + a_{yN} \sin (E + \omega)$$

$$e \sin E = a_{xN} \sin (E + \omega) - a_{yN} \cos (E + \omega)$$

$$r = a (1 - e \cos E)$$

$$\dot{r} = (\sqrt{\mu a}/r)\ e \sin E$$

$$r\dot{v} = (\sqrt{\mu a}/r) \sqrt{1 - e^2}$$

$$\cos u = \frac{a}{r} \left[\cos (E + \omega) - a_{xN} + a_{yN} \left(\frac{e \sin E}{1 + \sqrt{1 - e^2}} \right) \right]$$

$$\sin u = \frac{a}{r} \left[\sin (E + \omega) - a_{yN} - a_{xN} \left(\frac{e \sin E}{1 + \sqrt{1 - e^2}} \right) \right]$$

(7) Compute the components of the unit vectors $\underline{U}$ and $\underline{V}$:

$$\underline{U} = \underline{N} \cos i + \underline{N} \sin u$$

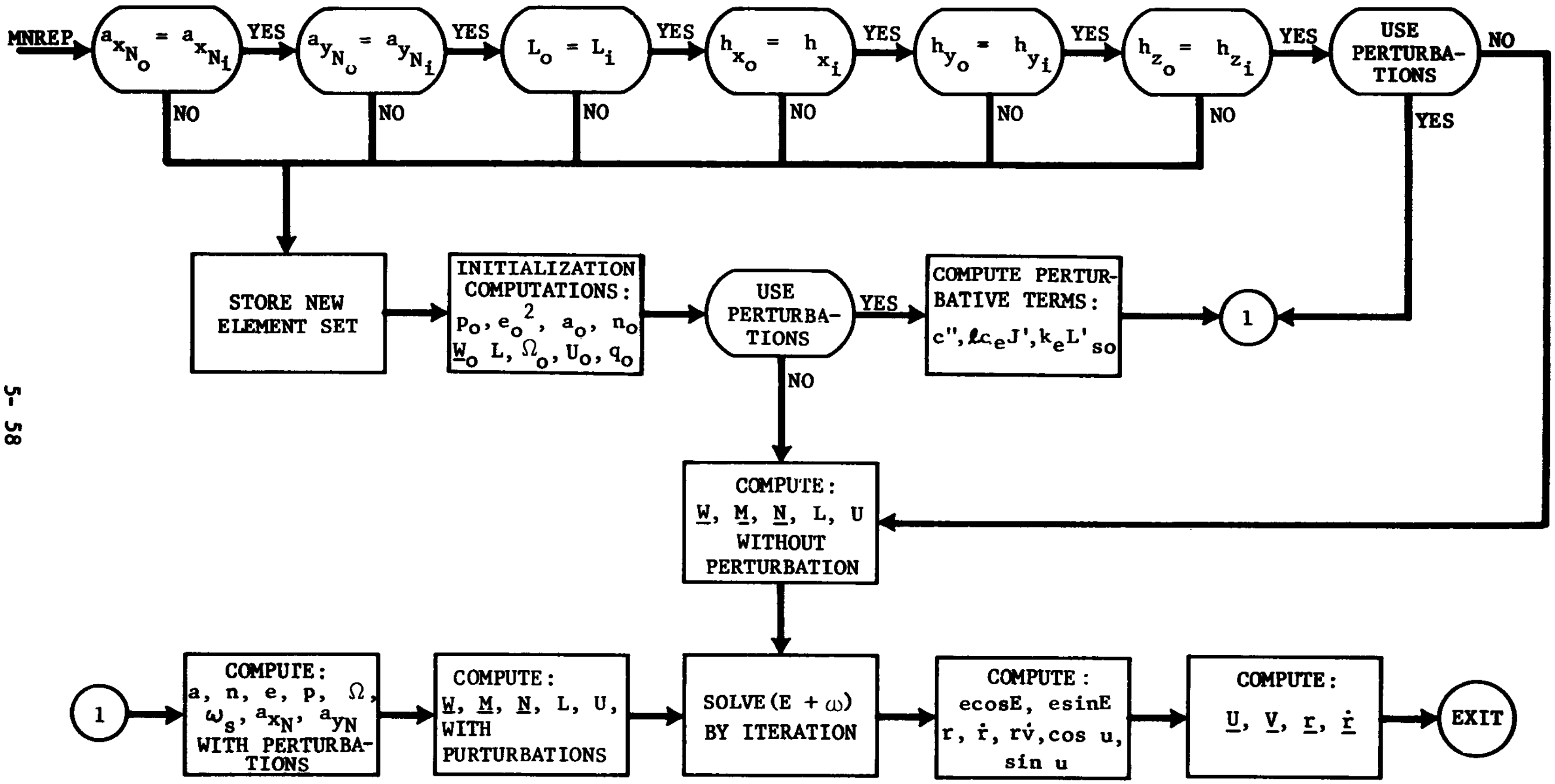
$$\underline{V} = -\underline{N} \sin u + \underline{M} \cos u$$

(8) The position and velocity vectors are then obtained directly:

$$\underline{r} = r\, \underline{U} \rightarrow x, y, z$$

$$\dot{r} = \dot{r}\, \underline{U} + r\dot{v}\, \underline{V} \rightarrow \dot{x}, \dot{y}, \dot{z}$$

MNREP


MNREP
$a_{x_{N_o}} = a_{x_{N_i}}$
YES
NO
$a_{y_{N_o}} = a_{y_{N_i}}$
YES
NO
$L_o = L_i$
YES
NO
$h_{x_o} = h_{x_i}$
YES
NO
$h_{y_o} = h_{y_i}$
YES
NO
$h_{z_o} = h_{z_i}$
YES
NO
USE PERTURBA- TIONS
YES
NO
STORE NEW ELEMENT SET
INITIALIZATION COMPUTATIONS: P_o, e_o^2, a_o, n_o $\underline{W}_o$ L, Ω_o, U_o, q_o
USE PERTURBA- TIONS
YES
NO
COMPUTE PERTUR- BATIVE TERMS: c'', $\ell c_e J'$, $k_e L'_{so}$
1
COMPUTE: $\underline{W}$, $\underline{M}$, $\underline{N}$, L, U WITHOUT PERTURBATION
1
COMPUTE: a, n, e, p, Ω, ω_s, a_{xN}, a_{yN} WITH PERTURBA- TIONS
COMPUTE: $\underline{W}$, $\underline{M}$, $\underline{N}$, L, U, WITH PURTURBATIONS
SOLVE (E + ω) BY ITERATION
COMPUTE: ecosE, esinE r, $\dot{r}$, r$\dot{v}$, cos u, sin u
COMPUTE: $\underline{U}$, $\underline{V}$, $\underline{r}$, $\dot{\underline{r}}$
EXIT

PURPOSE: To obtain a number randomly from a normal distribution.

CALL SEQUENCE:

```
JMP      NODN
TQM     (RESULT)
```

INPUTS: See Description

OUTPUTS: RESULT = Normal deviate (Floating point)

STORAGE REQUIREMENTS: 79 Cells

SUBROUTINES: Common: REDN

DESCRIPTION:

Since this subroutine uses REDN, it is necessary to initialize it as required (see REDN). Given a random number between 0 and 1, NODN will do a table look-up and interpolation to find the normal deviate.

OBCARD
(SEA)
11/30/62

PURPOSE: To control the conversion of a system observation card (internal format, Col 80 = 0) into OBLOC or COBLOC format.

CALL SEQUENCE: (1) For OBLOC: (2) For COBLOC:

```
JMP        OBCARD              JMP        OCCARD
```

INPUTS: See: OBSCRD

OUTPUTS: See: OBSCRD

STORAGE
REQUIREMENTS: 4 Cells

SUBROUTINES: Common: OOCARD, OBSCRD, COBCRD

DESCRIPTION:

OBCARD modifies subroutine OBSCRD to accept system observation cards; then transfers control to OBSCRD. Subroutine OOCARD is used in place of OBSERV. The OBCARD entrance is used in converting cards for OBLOC, and the OCCARD entrance is used in converting cards for COBLOC.

OBPCK
(SEA)
11/30/62

PURPOSE: To pack one simulated observation into OBLOC

CALL SEQUENCE: (1) To initialize: (2) Normal:

```
     JMP     IOBPCK                     JMP    OBPCK

                                 + 1H   JMP    (END)

                                 + 2H          (NORMAL)
```

INPUTS: LOBLOC = C/HLT, OBLOC; C/HLT, OBLOC + 3000

STAID
OTYPE
RANGE (See OBLOC format for contents of core locations)
ALPHA
DELTA
RODOT
T

OUTPUTS: One simulated observation in OBLOC (See OBLOC format)

END = OBLOC full (See Description)

STORAGE REQUIREMENTS: 48 Cells

SUBROUTINES: Common: PANT, GLOP, PAGECON

Program: OBPCKS, OBPCKA

DESCRIPTION:

It is necessary to "JMP IOBPCK" only before the first "JMP OBPCK". IOBPCK sets the beginning address of OBLOC. With every call of OBPCK this address is modified in order to pack the next observation successively; a program subroutine OBPCKS packs one observation. 500 observations may be stored in OBLOC. Observations in excess of 500 will be printed by program subroutine OBPCKA.

OBSCRD
(SEA)
11/30/62

PURPOSE: To control the conversion of a standard SPADATS observation card into OBLOC or COBLOC format.

CALL SEQUENCE: (1) For OBLOC: (2) For COBLOC:

```
(1)  For OBLOC:                   (2)  For COBLOC:

     JMP       OBSCRD                  JMP       COBCRD
```

INPUTS:

XR3 = Location of card image
EBLOC

(1) For OBLOC entrance:

LOBLOC = C/HLT, OBLOC; C/HLT, OBLOC + 3000

OBSSTB = C/HLT, OBLOC; C/HLT, 0 (initially)

(2) For COBLOC entrance:

LCOBLOC = C/HLT, COBLOC; C/HLT, COBLOC + 3000

COBSTB = C/HLT, COBLOC; C/HLT, 0 (initially)

OUTPUTS:

(1) For OBLOC entrance:

Left address of OBSSTB incremented by 6 and right address of LOCOBS set equal to it. OCOUNT incremented by 1.

(2) For COBLOC entrance:

Left address of COBSTB incremented by 6 and right address of LOCCOB set equal to it. COCOUNT incremented by 1.

Bit 46 set to 1 in location CARDS to indicate a successful read.

STORAGE REQUIREMENTS: 45 Cells

SUBROUTINES: Common: OBSERV, PANT, CARDER

DESCRIPTION:

OBSCRD and COBCRD are similar subroutines, differing only as to which block is used for output: OBLOC (Observation Block) or COBLOC (Contaminated Observation Block). It is necessary for element cards to be entered before observation cards. If not, the observation card is rejected and an error comment is printed. Standard SPADATS observation cards are not allowable for element types C and D; they will be rejected with an error comment if encountered. The output listed is obtained only if no error is encountered during card conversion. The maximum number of cards that will be accepted is 500; those in excess of 500 will be rejected with an error comment.

Also, see OBSERV.

OBSERV
(SEA)
11/30/62

PURPOSE: To convert a standard SPADATS observation card into OBLOC or COBLOC format.

CALL SEQUENCE:

```
TMA         (WORD)
JMP         OBSERV
JAZ         (ERROR)
```

INPUTS: WORD = C/HLT, (card address); C/HLT (Block address)
EBLOC

OUTPUTS: ERROR = Error on conversion
One observation entered in OBLOC or COBLOC

STORAGE REQUIREMENTS: 109 Cells

SUBROUTINES: Common: CARDER, PANT, SXRSRCH, BCDFD

DESCRIPTION:

This subroutine unpacks SPADATS observation cards into OBLOC or COBLOC. If an error is found, a comment will be printed and the normal exit will be taken with the A Reg. = 0.

OBSGET
(SEA)
11/30/62

PURPOSE: To retrieve one simulated observation from OBLOC.

CALL SEQUENCE:

```
     (1)  To initialize:                   (2)  Normal:
             JMP INITOBS                              JMP    OBSGET
                                                 +1H  JMP    (END)
                                                 +2H         (NORMAL)
```

INPUTS:

OBLOC
LOCOBS = C/HLT, OBLOC; C/HLT, 0
Termination indicator for OBLOC must be 48/0

OUTPUTS:

END = End of simulated observations
NORMAL A Reg = STAID
STAID
OTYPE
RANGE (See OBLOC format for contents of core locations)
ALPHA
DELTA
RODOT
T

STORAGE REQUIREMENTS: 20 Cells

SUBROUTINES: None

DESCRIPTION:

It is necessary to "JMP INITOBS" only before the first "JMP OBSGET". INITOBS sets the beginning address of OBLOC. With every call of OBSGET this address is modified to retrieve the next successive simulated observation. The maximum number of simulated observations is 500.

OOCARD
(SEA)
11/30/62

PURPOSE: To convert a system observation card (internal format, col 80 = 0) into OBLOC or COBLOC format

CALL SEQUENCE:

```
TMA     (WORD)
JMP     OOCARD
JAZ     (ERROR)
```

INPUTS: WORD = C/HLT, (Card address); C/HLT (Block address)

OUTPUTS: ERROR = Error on conversion
One observation entered in OBLOC or COBLOC

STORAGE REQUIREMENTS: 21 Cells

SUBROUTINES: Common: SXSRCH

SIMULATION ERROR ANALYSIS SYSTEM — PROGRAM DOCUMENT

PAGECON
(SEA)
11/30/62

PURPOSE: To provide a central control for printed output formats.

CALL SEQUENCE:

```
TMA         (WORD)
JMP         PAGECON
```

INPUTS: WORD = C/HLT, (N); C/HLT, 0, (N = number of lines of output)

OUTPUTS: A Reg = 0. Start of new page.

STORAGE
REQUIREMENTS: 7 Cells

SUBROUTINES: Common: HEAD, PANT, GLOP

DESCRIPTION:

This subroutine requires initialization by subroutine "PAGEINT" at the start of every job. It is recommended to use PAGECON at least after every 5 lines of output. To force a new page, set N = 70. The subroutine keeps a line count for the current output page. When the count is≥55, it will eject the page, put out a new heading, space 2 lines, and exit with the A Reg = 0.

PAGEINT
(SEA)
11/30/62

PURPOSE: To initialize HEAD and PAGECON.

CALL SEQUENCE: JMP PAGEINT

INPUTS: None

OUTPUTS: None

STORAGE REQUIREMENTS: 13 Cells

SUBROUTINES: Common: AKLOK, BCDCLK, HEAD

DESCRIPTION:

This subroutine initializes HEAD and PAGECON in the following manner:

1. Sets page number to 1
2. Uses Accounting Clock to set up BCD date and time of the start of run
3. Sets up the case ID and sequence number
4. Outputs the first page heading and sets the line count

NAME: PANT

PURPOSE: To edit output and control writing of output buffers on magnetic tape.

STORAGE
REQUIREMENTS: 806 Cells

SUBROUTINES: None

DESCRIPTION:

PANT edits all alphanumeric output data properly for the mode desired, accumulates each type in a separate buffer, writes out the buffer onto magnetic tape when it is full, and resets the buffer to the appropriate filler characters.

The three modes of output that PANT can edit are hard-copy (printer) teletype (TTY) and punched cards in the code-mode. The hard-copy is accumulated in PANT, BUFF; TTY in PANT.BBUF; and punched cards in PANT. PBUF. When processing the output tape off line, hard copy is in data select 0; punched cards in data select 2, and teletype in data select 4. The punched cards are in code-mode with the PCC (punched card controller)set to 10 words/card, 12 cards/block, and the "Sense Control Characters Switch" set to ignore.

PANT contains all its own output processing logic, including TIOs and skip check orders. Mode 3 writes are used which will normally handle parity and sprocket errors. In the case of block marks, PANT rewrites the block and indicates via a flex typeout that an improper block mark was encountered. Duplicate output may result in this case. Errors requiring operator action are indicated on the flexowriter. If PANT re-issues an order ten times without successfully writing the block, it will rewind the output tape with lockout (no wrapup), and type on the flex calling for a new output tape.

PANT has many entrances which are explained in detail on the following pages.

CALL SEQUENCE: JMP PANT.PAGE

INPUTS: None

OUTPUTS: Page eject on hard copy

DESCRIPTION:

It places a page eject in the hard copy output buffer followed by an end-of-line control character.

```
CALL SEQUENCE:   TMA     (WORD)
                 JMP     PANT.PINT

INPUTS:          WORD  =  C/HLT,N;C/HLT, TABLE      Form 1
                  or      N/XT23                    Form 2
```

In Form 1, the sum of TABLE + N is the location of a cell containing the desired tape number (O/T23) Eg TABLE + 5 contains O/5T23.

In Form 2, X = tape number desired

OUTPUTS: PANT initialized with buffers and counters all preset.

DESCRIPTION:

PINT initializes all IO and skip check orders to the tape number specified. It also sets all counters to the beginning address of the three output blocks (BUFF, BBUF, PBUF). It also sets the first word of each buffer to the proper data select (0 for printing, 2 for punched cards, 4 for TTY) and the remaining words in the buffer to the proper fill characters.

PINT will destroy any information contained in PANT's output buffers.

```
CALL SEQUENCE:  TMD       (WORD)
                JMP       PANT.PANT
```

INPUTS: WORD = C/HLT,N;C/HLT,BUFFER
N = No. of Words
BUFFER = Beginning address of alphanumeric data

OUTPUTS: Alphanumeric data in hard-copy output buffer or written on magnetic tape

DESCRIPTION:

PANT moves N words from Buffer to Buffer + N - 1 into the output buffer for hard-copy (PANT.BUFF). When PANT.BUFF contains 113 words or more, PANT writes it on magnetic tape, and resets it to fillers plus a data select word and an end of block word.

```
CALL SEQUENCE:  TMD     (WORD)
                JMP     PANT.TPANTA
             or
                JMP     PANT.TPANT
```

INPUTS: WORD = C/HLT, N;C/HLT, BUFFER
N = No. of words
BUFFER = Beginning address of alphanumeric data

PANT.TPANT assumes N = 15, BUFFER = GLOP.BUFF

OUTPUTS: Alphanumeric data on the teletype output buffer or written on magnetic tape.

DESCRIPTION:

PANT moves N words from Buffer to Buffer + N - 1 into the teletype output buffer (PANT.BBUF). When PANT.BBUF contains 127 words plus a data select word, PANT writes it on magnetic tape and resets the buffer to letters characters (37_8).

If the user wants to use the same GLOP.BUFF output which has just been put on hard copy for teletype output, he should jump to PANT.TPANT. This entrance assumes N = 15, BUFFER = GLOP.BUFF.

In order to minimize the number of characters in a line, TPANT will recognize a 32_8 (filler character) as the end of the teletype line and convert it to two carriage returns and a line feed (555575_8). If the user does not have a 32 in his output data, TPANT will put the two carriage returns and line feed after the last character of the last word, (8th character of Buffer + N - 1). When using TPANT, one should always output the 32_8 character immediately following the last column used to minimize the amount of teletype tape punched and the amount of transmission time necessary to send the message. This is especially important when using GLOP.BUFF either via GLOP.TGLOP or PANT.TPANT.

CALL SEQUENCE: TMD (WORD)
JMP PANT.PPANT

INPUTS: WORD = C/HLT, N;C/HLT, BUFFER
N = No. of words
BUFFER = Beginning address of alphanumeric data

OUTPUTS: Alphanumeric data in the punched card output buffer or written on magnetic tape in proper format for punching.

DESCRIPTION:

PPANT moves N words from Buffer to Buffer + N - 1 into the punched card output buffer (PANT.PBUF). It then sets the next "10-N" locations equal to blanks. If N is greater than 10, PPANT only accepts the first 10 words of BUFFER. When PANT.PBUF has 12 card images (or 120 words plus a data select word), it is written on magnetic tape and reset to fillers (32_8). Cards are punched with the PCC (punch card controller) set at 10 words/card, 12 cards/block, and sense control characters to ignore.

```
CALL SEQUENCE:  JMP         PANT.SPACE
            or
                TMA         (WORD)
                JMP         PANT.SPACER

INPUTS:         WORD = C/HLT, N
                       N = Number of spaces desired

                PANT.SPACE assumes N = 1

OUTPUTS:        Spaces on hard copy
```

DESCRIPTION:

Puts N end-of-line characters in the output buffer in PANT (PANT.BUFF).

```
CALL SEQUENCE:  JMP        PANT.TSPACE
              or
                TMD        (WORD)
                JMP        PANT.TSPACER
```

INPUTS: WORD = C/HLT, N
N = Number of spaces desired

PANT.SPACE is the special case where N is assumed = 1

OUTPUTS: One or more line feeds on teletype output

DESCRIPTION:

The number of line feeds specified are output with a double carriage return, preceding the last line feed. If N = 0, no output is done and a normal return is executed.

CALL SEQUENCE: JMP PANT.FINISH

INPUTS: None

OUTPUTS: All hard copy output on magnetic tape with a conditional stop character.

DESCRIPTION:

If the hard-copy output buffer is partially full, PANT writes it on magnetic tape and resets the buffer to fillers. Then, in all cases, a conditional stop is inserted and the output buffer is written on tape and reset to filler characters.

CALL SEQUENCE: JMP PANT.TFIN

INPUTS: None

OUTPUTS: All TTY output on magnetic tape with a conditional stop character.

DESCRIPTION:

If the TTY output buffer is partially full, it is written on magnetic tape and reset to letters characters. Then, in all cases, a conditional stop is inserted and the output buffer is written on tape and reset to letters characters.

CALL SEQUENCE: JMP PANT.PFIN

INPUTS: None

OUTPUTS: All punched card output on magnetic tape with a conditional stop character.

DESCRIPTION:

If the punched card output buffer is partially full, PANT writes it on magnetic tape and resets it to fillers. Then, in all cases, a conditional stop character is inserted and the output buffer is written on tape and reset to filler characters.

```
CALL SEQUENCE:  JMP    PANT.ALLFIN    (Conditional stops)
             or
                JMP    PANT.XFIN      (Absolute stops)
```

INPUTS: None

OUTPUTS: All output data on magnetic tape with absolute or conditional stops.

DESCRIPTION:

This is equivalent to a jump to PANT.FINISH, PANT.TFIN, PANT.PFIN. If PANT.XFIN is used, absolute, rather than conditional stops, will be output.

PURPOSE:	Output image of Col. 25 to 80 of PRINT card to hard-copy.
CALL SEQUENCE:	JMP PRINT
INPUTS:	XR3 = card location
OUTPUTS:	XR3 = card location + 3 Alphanumeric data output.
STORAGE REQUIREMENTS:	4-1/2 Cells
SUBROUTINES:	Common: CARDER, PANT, PAGECON

DESCRIPTION:

Outputs the 7 words from card location + 3 to card location + 9 to hard-copies, then outputs one space.

SIMULATION ERROR ANALYSIS SYSTEM **PROGRAM DOCUMENT**

REDN
(SEA)
11/30/62

PURPOSE: To generate a random number.

CALL SEQUENCE:

```
     (1)  To initialize:                    (2)  Normal:
               TMA       (WORD)                        JMP       REDN
               JMP       REDNI                         TAM       (RESULT1)
                                                       TQM       (RESULT2)
```

INPUTS: WORD = First random number of the form 4K + 1, K is a positive integer (fixed point)

OUTPUTS: RESULT1 = Fixed point random number
RESULT2 = Floating point random number

STORAGE REQUIREMENTS: 9 Cells

SUBROUTINES: None

DESCRIPTION:

It is necessary to initialize only once for each case or for each time a new beginning random number is desired. REDN saves the current random number and uses it to generate the next random number.

SIMULATION ERROR ANALYSIS SYSTEM **PROGRAM DOCUMENT**

SCAN
(SYS)
11/30/62

PURPOSE: To extract a variable field on an input card.

CALL SEQUENCE:

(1) For first field:
```
JMP      SCAN
```

(2) For successive fields:
```
JMP      SCANON
```

INPUTS: Input card

OUTPUTS:

W1
W2 } BCD field, blanks suppressed, right justified with leading zeros.

1CHARCT = Left address = character count

A Reg = break character

STORAGE REQUIREMENTS: Part of SYS

SUBROUTINES: None

DESCRIPTION:

This subroutine will extract variable fields for columns 25-80; maximum allowable field length is 16 BCD characters. To extract the first field, "JMP SCAN"; for successive fields, "JMP SCANON". The subroutine will search field, suppressing blanks and storing valid characters into core locations W1 and W2 until a break character is found (,$./;). The character count is kept in the left address of core location 1CHARCT. Upon return, the A register will contain the break character, right justified with leading zeros.

If blanks are desired, precede "JMP SCANON" by a "JMP GETBL". If blanks are desired on the first field, move core location COL25 to core location SCANC, "JMP GETBL", and begin with "JMP SCANON" rather than "JMP SCAN".

SCARD
(SEA)
11/30/62

PURPOSE: To convert a sensor card into SBLOC format.

CALL SEQUENCE:

```
TMA        (WORD)
JMP        SCARD
JAZ        (ERROR)
           (NORMAL)
```

INPUTS: WORD = C/HLT, (Card address); C/HLT, (Block address)

OUTPUTS: ERROR = Error on conversion

One sensor entered in SBLOC.

STORAGE REQUIREMENTS: 60 Cells

SUBROUTINES: Common: SFXFLT, SFXFLTA, CARDER, PANT

Program: SCONV

Philco: FSIN, FCOS, FSQRT

DESCRIPTION:

This subroutine unpacks sensor cards into SBLOC. It also computes X/cos θ and Z by using program subroutine SCONV; these values are also stored in SBLOC. If an error is found, a comment will be printed and the normal exit will be taken with the A Reg = 0.

SDKLOK
(SEA)
11/30/62

PURPOSE: To convert floating point days to a BCD date

CALL SEQUENCE:

```
TMA         (WORD)
JMP         SDKLOK
TAM         (RESULT1)
TQM         (RESULT2)
```

INPUTS: WORD = Floating point days since beginning of year

OUTPUTS: RESULT1 = △△YYMMDD (6-Bit BCD)

RESULT2 = Fraction of days (floating point)

STORAGE REQUIREMENTS: 122 Cells

SUBROUTINES: None

PURPOSE:	To control the conversion of a sensor card into SBLOC format
CALL SEQUENCE:	JMP SENCARD

INPUTS:	XR 3 = Location of Card image
	LSBLOC = C/HLT, SBLOC; C/HLT, SBLOC + 600
	SENSTB = C/HLT, SBLOC; C/HLT, 0 (Initially)

OUTPUTS:	The left address of SENSTB is incremented by 6 and the right address of LOCSEN is set equal to it. SCOUNT is incremented by 1. Bit 2 of location CARDS is set to 1.
STORAGE REQUIREMENTS:	11 Cells
SUBROUTINES:	Common: CARDER, SCARD

DESCRIPTION:

The output listed is obtained only if no error is encountered during the card conversion. The maximum number of cards that will be accepted is 100. Those in excess of 100 will be rejected with an error comment.

SIMULATION ERROR ANALYSIS SYSTEM — PROGRAM DOCUMENT
SENGET
(SEA)
11/30/62

PURPOSE: To retrieve sensor data from SBLOC.

CALL SEQUENCE:

(1) To initialize:

```
        JMP    INITSEN
```

(2) To retrieve sensors sequentially:

```
        CA
        JMP    SENGET
+ 1H    JMP    (END)
+ 2H           (NORMAL)
```

(3) To retrieve a specific sensor:

```
        TMA    (WORD)
        JMP    SENGET
+ 1H    JMP    (ERROR)
+ 2H           (NORMAL)
```

INPUTS:

LOCSEN = C/HLT SBLOC; C/HLT, 0
Termination indicator for SBLOC must be 48/0
SCOUT = C/HLT, N; C/HLT, 0 N = No. of Sensors in SBLOC
WORD = ΔΔΔΔΔ NNN, N = Sensor Number
SBLOC

OUTPUTS:

END = End of SBLOC ERROR = Sensor not in SBLOC

XLAMBA
PHIRD
OALT
XOVCT (See SBLOC format for contents of core locations)
CAPZ
STAID

STORAGE REQUIREMENTS: 27 Cells

SUBROUTINES: None

DESCRIPTION: This subroutine may be used in two ways: To retrieve a specific sensor or to retrieve the next sequential sensor. To retrieve a specific sensor use call sequence (3); no initialization is required. If retrieving sequential sensors, it is necessary to "JMP INITSEN" only before the first use of call sequence (2). INITSEN sets the beginning address of SBLOC. With every call of SENGET (2), this address is modified to retrieve the next sensor. The maximum number of sensors is 100.

SEPSUB
(SEA)
11/30/62

PURPOSE:	To separate a floating point number into its integral and fractional parts.	
CALL SEQUENCE:	TMA	(WORD)
	JMP	SEPSUB
INPUTS:	WORD = Floating point number	
OUTPUTS:	A Reg. = Integer (F)	
	Q Reg. = Fraction (F)	
STORAGE REQUIREMENTS:	18 Cells	
SUBROUTINES:	None	

SETJC
(SEA)
11/30/62

PURPOSE: To set the mode of the MNREP subroutine.

CALL SEQUENCE:

```
TMA          (WORD)
JMP          SETJC
```

INPUTS:

$$\text{WORD} = \left\{\begin{array}{l} \left.\begin{array}{l}00000000\\0000000\Delta\end{array}\right\} \text{Allow perturbations} \\ \left.00000001\right\} \text{Suppress perturbations}\end{array}\right.$$

OUTPUTS:

$$\text{XJGRCF} = \left\{\begin{array}{l}0\\1.2071762 \times 10^{-4}\end{array}\right\} \begin{array}{l}\text{Suppress perturbations}\\\text{Allow perturbations}\end{array}$$

STORAGE REQUIREMENTS: 4 Cells

SUBROUTINES: None

DESCRIPTION:

The core location XJGRCF is set as a parameter for the MNREP subroutine to determine whether the two-body or the simplified general perturbations approach is used for predicting orbital position and velocity.

SIMULATION ERROR ANALYSIS SYSTEM **PROGRAM DOCUMENT**
SFXFLT
(SEA)
11/30/62

PURPOSE: To convert a BCD number to a floating point number.

CALL SEQUENCE:

```
(1)   If ≤ 8 BCD characters:         (2)   If > 8 and ≤ 16 BCD
      TMQ     (WORD)                       characters:
      JMP     SFXFLTA                      TMA      (WORD 1)
      JAZ     (ERROR)                      TMQ      (WORD 2)
      TAM     (RESULT)                     JMP      SFXFLT
                                           JAZ      (ERROR)
                                           TAM      (RESULT)
```

INPUTS: WORD = ≤ 8 BCD Character number

WORD 1
WORD 2 > 8 and ≤ 16 BCD Character number

OUTPUTS: RESULT - floating point number

STORAGE REQUIREMENTS: 110 Cells

SUBROUTINES: None

DESCRIPTION:

The input may be of the following forms:

±xx. xx

±xx. xx ± NN

±xx. xx E ± NN

All blanks are ignored. Leading zeros are ignored; trailing zeros are significant.

SIGGET
(SEA)
11/30/62

PURPOSE: To retrieve one set of standard deviations and biases from SGBLOC.

CALL SEQUENCE:

```
     (1)  To initialize                 (2)  To retrieve sequentially:
                JMP   INITSIG                          CA
                                                       JMP   SIGGET
                                             + 1H      JMP   (END)
                                             + 2H            (NORMAL)

     (3)  To retrieve specifically:
                TMA   (WORD)
                JMP   SIGGET
          + 1H  JMP   (ERROR)
          + 2H        (NORMAL)
```

INPUTS:

LOCSIG = C/HLT, SGBLOC: C/ HLT, 0
Termination indicator for SGBLOC must be 48/0
SGCOUNT = C/HLT, N; C/HLT, 0 N = Number of sets in SGBLOC
WORD = △△△△△ NNN, N = Sensor Number

OUTPUTS:

```
END = End of SGBLOC    ERROR = Set not in SGBLOC
STAID      BIASRHO
SIGRHO     BIASALP
SIGALP     BIASDLT
SIGDLT     BIASRRT  (See SGBLOC Format for contents of
SIGRRT              core locations)
```

STORAGE REQUIREMENTS: 24 Cells

SUBROUTINES: None

DESCRIPTION:

This subroutine may be used in two ways: To retrieve one set of standard deviations and biases for a specific sensor or for the next sequential sensor. To retrieve a set for a specific sensor, use call sequence (3); no initialization is required. If retrieving a set for sequential sensors, it is necessary to "JMP INITSIG" only before the first use of call sequence (2). INITSIG sets the beginning address of SGBLOC. With every call of SIGGET (Call sequence 2) this address is modified to retrieve the next set of deviations and biases. The maximum number of sets is 100.

PURPOSE: To convert a Simulation Parameter Card into IBLOC format.

CALL SEQUENCE:

```
TMA     (WORD)
JMP     SPARM
JAZ     (ERROR)
```

INPUTS: WORD = C/HLT, (Card address); C/HLT, (Block address)

OUTPUTS: ERROR = Error on conversion
One set of simulation parameters entered in IBLOC - see IBLOC format.

STORAGE REQUIREMENTS: 93 Cells

SUBROUTINES: Common: CARDER, SFXFLTA, SFXFLTA

DESCRIPTION:

The subroutine converts a Simulation Parameter Card into IBLOC format. If an error is found, the card will be rejected with an error comment; the normal exit will be taken with the A Reg = 0.

SIMULATION ERROR ANALYSIS SYSTEM PROGRAM DOCUMENT
SPOUT 1
(SEA)
11/30/62

PURPOSE: To output results of a distribution of the delta classical elements.

CALL SEQUENCE: JMP SPOUT 1

INPUTS:
VQ Table (from STOB)
HBLOC (from STOB)
STCOUNT (counter in STOB)

OUTPUTS:
Label
S1, S2, NS, I
σ, μ, e_{min}, e_{max}, Δ_e, e_o
Histogram

STORAGE REQUIREMENTS: 222 Cells

SUBROUTINES:
Common: PANT, GLOP, PAGECON
Program: FDOUT

DESCRIPTION:

This is a subroutine used by STOB to output the results of a distribution of the delta classical elements. If C_o is not corrected, it will not be processed by STOB and SPOUT 1 will print a comment. FDOUT is a program subroutine to output the histogram; it is also used by SPOUT 2.

SPOUT 2
(SEA)
11/30/62

PURPOSE: To output results of a distribution of the delta position and velocity elements.

CALL SEQUENCE: JMP SPOUT2

INPUTS:

OPOUT1	
OPOUT2	
VQ Table	(See STOB)
HBLOC	(See STOB)
OCOUNT, STCOUNT	(Counters in STOB)
CASETYP	(From ERP)
UNITEST	(See Description)

OUTPUTS: This subroutine has several output options:

(1) CASETYP $\neq$ 2
OPOUT1 = 0, Print label epoch, min. since epoch, μ, σ, e_{min}, e_{max}
OPOUT1 $\neq$ 0, Print label, epoch, min. since epoch, μ, σ, e_{min}, e_{max}, histogram.
OPOUT2 = 0, Punch label, epoch, min. since epoch μ, σ, e_{min}, e_{max}
OPOUT2 $\neq$ 0, None

(2) CASETYP = 2 (Quad. Analysis)
STPBLOC will be built, while
OPOUT2 will be tested but OPOUT1 will be bypassed when STPBLOC is built. Label, μ, σ, e_{min}, e_{max} will be printed.

STORAGE REQUIREMENTS: 161 Cells

SUBROUTINES: Common: PANT, GLOP, PAGECON, SPOUT1
Program: STVQ, VQPUNCH, FDOUT (See SPOUT1), STOUT3

DESCRIPTION:

This is a subroutine to be used by STOB to output the results of a distribution of the delta position and velocity elements. When this subroutine is used bv STOB, STOB will expect
UNITEST = 0/5T15 for DPVBLOC
0/15T15 for STPBLOC
as additional input.

SIMULATION ERROR ANALYSIS SYSTEM **PROGRAM DOCUMENT**

STAB
(SEA)
11/30/62

PURPOSE: To issue a single tab or double tab to the flexowriter.

CALL SEQUENCE:

```
     (1)   For single tab:              (2)   For double tab:
              JMP    STAB                        JMP    DBLTAB
```

INPUTS: None

OUTPUTS: Tab characters transmitted to the flexowriter.

STORAGE REQUIREMENTS: 4 Cells

SUBROUTINES: None

STDEV
(SEA)
11/30/62

PURPOSE: To convert a standard deviation and bias card (M card) into SGBLOC format.

CALL SEQUENCE:

```
TMA     (WORD)
JMP     STDEV
JAZ     (ERROR)
```

INPUTS: WORD = C/HLT (Card address); C/HLT, (Block address)

OUTPUTS: ERROR = Error on conversion
One set of standard deviations and biases entered in SGBLOC

STORAGE REQUIREMENTS: 45 Cells

SUBROUTINES: Common: SFXFLTA, CARDER

DESCRIPTION:

This subroutine converts an M card into SGBLOC format. If an error is found, the card will be rejected with an error comment; the normal exit will be taken with the A Reg = 0.

STLC
(SEA)
11/30/62

PURPOSE: To retrieve three astrodynamical constants which vary with the year.

CALL SEQUENCE:

```
TMQ      REFYEAR
JMP      STLC
```

INPUTS: REFYEAR = 0000000Y, Y = Significant year digit in BCD
Option: 0000NNNN, N = Full year, with program extracting the year digit

OUTPUTS:
THGRO = θ_o (Radians)
XLSUNO = $L_{\odot}^{o}$ (Radians)
C3 = $-C_3$ (Radians)

STORAGE REQUIREMENTS: 44 Cells

SUBROUTINES: None

DESCRIPTION:

This subroutine is only usable for the years 1957 - 1966, as it interprets only the last digit to retrieve the constants, i.e., 8 → 1958, 1 → 1961.

PURPOSE: To statistically process any sample size of a set of variant quantities.

CALL SEQUENCE:

```
TIJ     (WORD)
TJM     ST11
JMP     STOB
```

INPUTS:

WORD = Name of output routine
OCOUNT = 00000000 (Counter)
TEST NS - If NS > N, make NS = N
VQ table must be filled in as indicated by asterisks (STOB will fill in other values) - See attached format.
TN = Total no. of sets of quantities (D/T47)
Lable Table
Block of TN sets of quantities

OUTPUTS:

μ, Σ_e
σ, Σ_e^2
e_{min}, Δe,
e_{max}, e_o , and frequency distribution (HBLOC) also see SPOUT1 and SPOUT2

STORAGE REQUIREMENTS: 377 Cells

SUBROUTINES: Common: SEPSUB, PAGECON, PANT
Program: ST13

DESCRIPTION:

This subroutine processes any sample size of a family of variant quantities and produces a statistic characteristic of the set of quantities. The statistic is given in terms of the mean, standard deviation, minimum, maximum, and frequency distribution.

The label table is a BCD list of names describing the quantities to be processed; BCD names should be centered in the required 3 words/name (This is necessary for printing purposes). Also the names should correspond sequentially to the quantities of the set in the block.

The block format is fixed by sets, i.e., (x_1, y_1, z_1, x_2, y_2, z_2, . . . x_{TN}, y_{TN}, z_{TN}). In this example there are "TN" sets of quantities which must be in the floating point format.

DESCRIPTION: (continued)

Not all quantities in a set must be processed. (VQ + 21) must be the number of sequential quantities which will be processed. For example, if there are 12 quantities in each set, and the 1st, 2nd, 4th, and 6th are to be processed, the cells would be set as follows:

VQ + 18 = D/12 T15
VQ + 21 = D/6 T15
VQ + 24 = 1/110101 T5

In such a case, the label table must contain labels describing the 1st 6 entries in the set, although it is desired to suppress the 3rd and 5th quantities. This is necessary because a comment is printed pertaining to the suppressed quantities: (LABEL) - "HISTOGRAM NOT PROCESSED". If all 12 are to be processed, the label table should have 12 labels and the cells would be set as follows:

VQ + 18 = D/12 T15
VQ + 21 = D/12 T15
VQ + 24 = 12/1T11

The subroutine will first move the appropriate label to VQ table, and then compute or find the following values:

(1) If NS = 0, then NS = TN
If I = 0, then I = 1/2 (TN) + 1

(2) Σe, Σe^2 for NS quantities

(3) If NS > 1, then $\mu = \frac{\Sigma e}{NS}$, $\sigma = \sqrt{\frac{1}{NS-1} \left| \Sigma e^2 - \frac{(\Sigma e)^2}{NS} \right|}$

If NS = 1, then $\mu = \Sigma e$, $\sigma = 0$

(4) If S1 ≠ 0, then $e_o = \mu - S1\sigma$ $\Delta e = \frac{(S1 + S2)\sigma}{I}$

If SI = 0, then $e_o = \max\left[(\mu - S2\sigma), 0\right]$, $\Delta e = \frac{\mu + S2\sigma - e_o}{I}$

(5) Σe, Σe^2, e_{min}, e_{max} of TN quantities

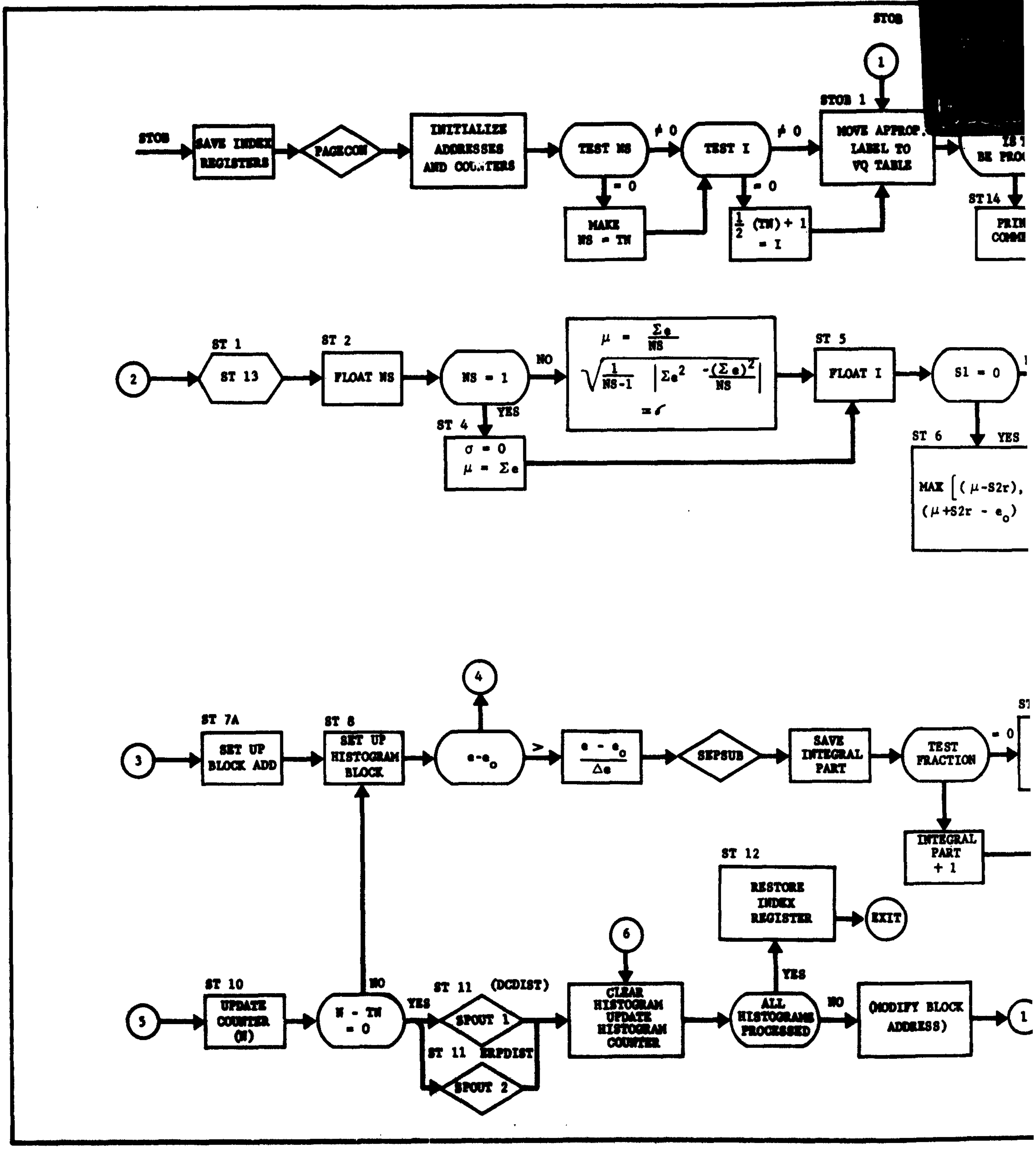
The subroutine will then examine each quantity and determine the cell in the distribution in which it belongs. If the quantity is less than or equal to e_o, it will be considered equal to e_o and therefore entered in

DESCRIPTION: (continued)

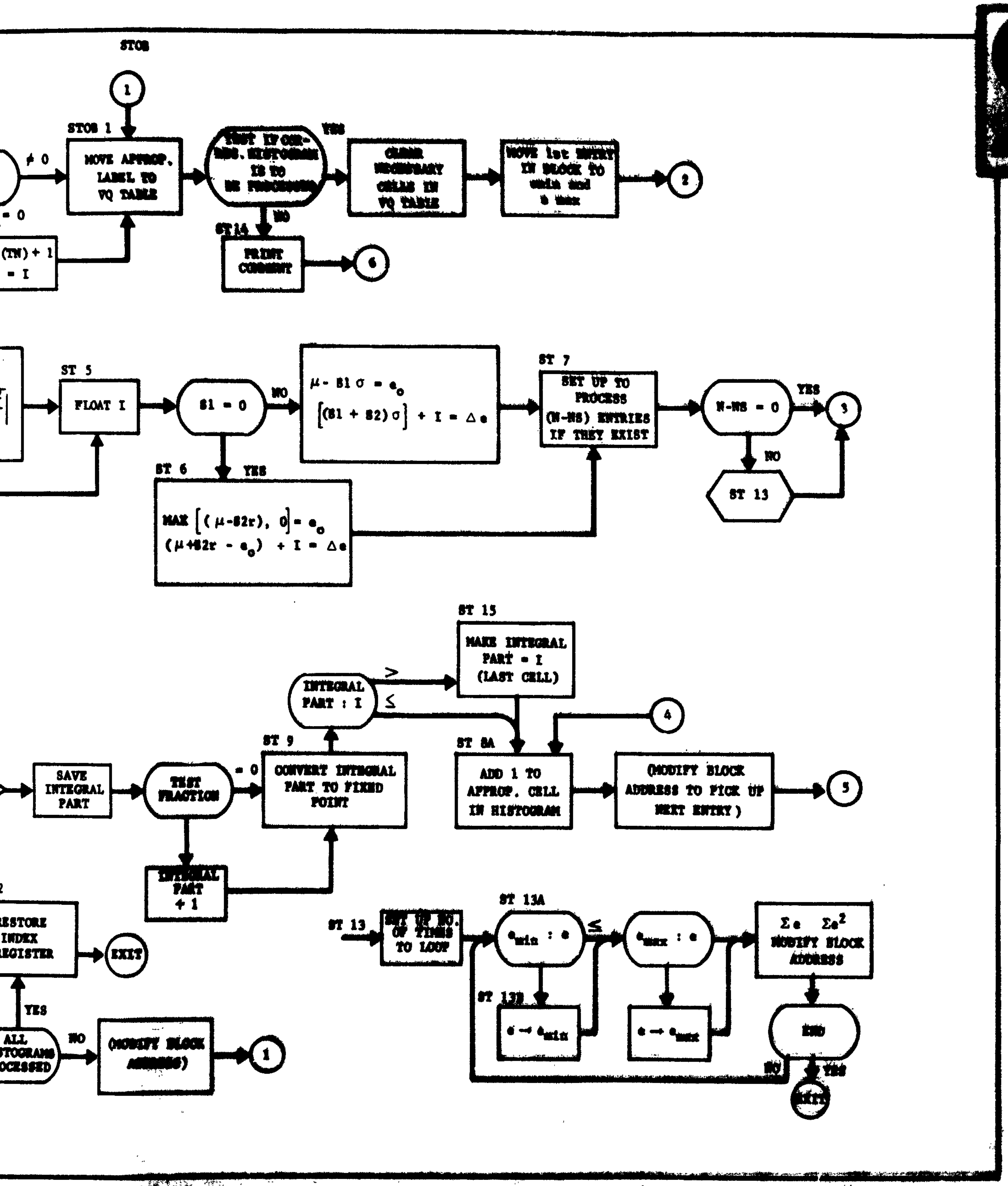
the first cell, otherwise the cell is determined. The integral part of

$$\frac{e - e_o}{\Delta e} + \begin{cases} 0, \text{ if fractional part} = 0 \\ 1, \text{ if fractional part} \neq 0 \end{cases}$$

If the result of this equation is greater than I, the quantity will be considered as belonging in the Ith cell. The distribution is located in HBLOC to HBLOC + I - 1, HBLOC + 99 being the maximum. The subroutine will go to the specified output subroutine after each quantity has been processed as none of the results are retained in core. After processing all quantities, the subroutine will exit. SPOUT 1 is the output routine for the DC distribution, and SPOUT 2 for the error prediction distribution.


STOB
1
STOB 1
STOB
SAVE INDEX REGISTERS
PAGECON
INITIALIZE ADDRESSES AND COUNTERS
TEST NS
≠ 0
= 0
MAKE NS = TN
TEST I
≠ 0
= 0
$\frac{1}{2}$ (TN) + 1 = I
MOVE APPROP. LABEL TO VQ TABLE
IS T
BE PROC
ST14
PRIN
COMM
2
ST 1
ST 13
ST 2
FLOAT NS
NS = 1
NO
YES
ST 4
$\sigma = 0$
$\mu = \Sigma e$
$\mu = \frac{\Sigma e}{NS}$
$\sqrt{\frac{1}{NS-1} \left| \Sigma e^2 - \frac{(\Sigma e)^2}{NS} \right|} = \sigma$
ST 5
FLOAT I
S1 = 0
ST 6
YES
MAX $[(\mu - S2r)$,
$(\mu + S2r - e_o)$
4
3
ST 7A
SET UP BLOCK ADD
ST 8
SET UP HISTOGRAM BLOCK
$e - e_o$
>
$\frac{e - e_o}{\Delta e}$
SEPSUB
SAVE INTEGRAL PART
TEST FRACTION
= 0
ST
INTEGRAL PART + 1
ST 12
RESTORE INDEX REGISTER
EXIT
6
5
ST 10
UPDATE COUNTER (N)
N - TN = 0
NO
YES
ST 11
(DCDIST)
SPOUT 1
ST 11
ERPDIST
SPOUT 2
CLEAR HISTOGRAM UPDATE HISTOGRAM COUNTER
ALL HISTOGRAMS PROCESSED
YES
NO
(MODIFY BLOCK ADDRESS)
1

[redacted]

STOB
1
STOB 1
≠ 0
MOVE APPROP. LABEL TO VQ TABLE
= 0
(TN) + 1
= I
TEST IF CORRES. HISTOGRAM IS TO BE PROCESSED
YES
CLEAR NECESSARY CELLS IN VQ TABLE
MOVE 1st ENTRY IN BLOCK TO emin and e max
2
NO
ST 1A
PRINT COMMENT
6
ST 5
FLOAT I
S1 = 0
NO
μ - S1 σ = e_o
[(S1 + S2) σ] + I = Δ e
ST 7
SET UP TO PROCESS (N-NS) ENTRIES IF THEY EXIST
N-NS = 0
YES
3
NO
ST 13
ST 6
YES
MAX [(μ-S2r), 0] = e_o
(μ+S2r - e_o) + I = Δ e
ST 15
MAKE INTEGRAL PART = I (LAST CELL)
INTEGRAL PART : I
>
≤
4
ST 9
ST 8A
SAVE INTEGRAL PART
TEST FRACTION
= 0
CONVERT INTEGRAL PART TO FIXED POINT
ADD 1 TO APPROP. CELL IN HISTOGRAM
(MODIFY BLOCK ADDRESS TO PICK UP NEXT ENTRY)
5
INTEGRAL PART + 1
2
RESTORE INDEX REGISTER
EXIT
YES
ALL HISTOGRAMS PROCESSED
NO
(MODIFY BLOCK ADDRESS)
1
ST 13
SET UP NO. OF TIMES TO LOOP
ST 13A
e_min : e
≤
e_max : e
Σ e   Σ e²
MODIFY BLOCK ADDRESS
ST 13B
e → e_min
e → e_max
END
NO
YES
EXIT

Location	Notation	Description	Format
VQ	Label	Maximum of 24 BCD characters	BCD
+ 1	Label	Describing the quantity being	BCD
+ 2	Label	processed	BCD
* + 3	S1	Standard deviation below the mean	F
* + 4	S2	Standard deviation above the mean	F
* + 5	NS	No. of quantities used to compute range of distribution	D/T47
+ 6	NS	No. of quantities used to compute range of distribution	F
* + 7	Addresses	C/HLT, (LABEL TABLE); C/HLT, (BLOCK ADDRESS)	
* + 8	I	No. of cells in the frequency distribution	D/T47
+ 9	e_o	Lower end of range of distribution	F
+10	Δe	Width of each cell in distribution	F
+11	e_{min}	Smallest quantity	F
+12	e_{max}	Largest quantity	F
+13	μ	Mean of NS quantities	F
+14	σ	Standard deviation of NS quantities	F
+15	N	Counter (incremented by 1 until = TN)	D/T47
+16	Σe	Sum of TN quantities	F
+17	Σe^2	Sum of squares of TN quantities	F
* +18		Block retrieval interval (7, 8, or 24)	D/T15
+19	NS	(Same as VQ + 5)	D/T39
+20	I	(Same as VQ + 8)	F
* +21		No. of sequential quantities to be processed (6, 7, 8, 24)	D/T15
+22		Current cell in distribution to be incremented	F
+23		(Same as VQ + 7 - but modified)	
* +24		Bits (left justified) = quantities to process {1 = yes 0 = no}	

STRIP
(SYS)
11/30/62

PURPOSE: To delete blanks from a BCD word.

CALL SEQUENCE:

```
TMQ    (WORD)
JMP    STRIP
TAM    (RESULT)
```

INPUTS: WORD = 8 characters possibly containing blanks
(example: ΔT Δ ES Δ TΔ)

OUTPUTS: RESULT = Word, all blanks removed, right justified with leading zeros.
(Example: 0000 TEST)

STORAGE REQUIREMENTS: Part of SYS

SUBROUTINES: None

PURPOSE: To aid in processing input cards

CALL SEQUENCE:

```
TMA     (WORD)
JMP     SXSRCH
JAZ     (ERROR)
```

INPUTS: WORD = C/HLT, 0; C/HLT, (Address of format word list)
See description for format word list.

OUTPUTS: See Description

STORAGE REQUIREMENTS: 91 Cells

SUBROUTINES: Common: CARDERB, PAGECON, SFXINT, SFXFLT, PANT.

DESCRIPTION:

The format word list is a set of parameters used to unpack, convert, and store specified fields (max. - 16 col) on input cards. A format word has the form:

C/(Command 1), (Address 1); C/(Command 2), (Address 2)

$$\text{Command 1} = \begin{cases} \text{HLT, if alphanumeric conversion desired} \\ \text{JBT, if fixed integer conversion desired} \\ \text{ICOZ, if floating point conversion desired} \end{cases}$$

Address 1 = P * 128 + F, P = last column in field, $0 < P \leq 80$
F = no. of columns in field, $0 < F \leq 16$

$$\text{Command 2} = \begin{cases} \text{TMA, for all format words except the last} \\ \text{CAM, if last format word in list} \end{cases}$$

Address 2 = Location in which converted field is to be stored

Alphanumeric conversion leaves data right justified, preceded by blanks. If $F \leq 8$, only one word will be stored in Address 2. If $F > 8$, two words will be stored in Address 2 and Address 2 + 1.

Fixed Integer Conversion leaves a fixed binary integer T47 in Address 2. The conversion will terminate on a decimal point, therefore △△75.62, △△△ 75.0, and △△△△△ 75 will all convert as D/75. Preceeding + or - signs are allowed.

DESCRIPTION: (continued)

Floating point Conversion leaves a floating point number in Address 2 with one exception: If the field specified is all blanks, Address 2 will contain OCTAL 0. Input can be any of the following formats.

±XXXX.XXX

±XXXXX

±XX.XXXX±XXX

±XXX.XXXE±XXX

All preceding, imbedded, and following blanks will be suppressed in both fixed and floating conversion. Preceding zeros will also be suppressed. Therefore, 0 + 10 will convert as D/10 or F/10. Depending on the type of conversion specified.

If an error occurs on any field, SXSRCH will print the diagnostics before returning control. The entire card is always converted.

SIMULATION ERROR ANALYSIS SYSTEM

PROGRAM DOCUMENT
VECARD
(SEA)
11/30/62

PURPOSE: To convert a pair of variant element cards into VEBLOC format.

CALL SEQUENCE:

```
TMA    (WORD1)
TMQ    (WORD2)
JMP    VECARD
JAZ    (ERROR)
```

INPUTS: WORD1 = C/HLT, (Card 1 address); C/HLT, 0
WORD2 = C/HLT, DEBLOC; C/HLT, VEBLOC

OUTPUTS: ERROR = Error on conversion
One set of N M elements in VEBLOC
One set of classical elements in DEBLOC

STORAGE REQUIREMENTS: 56 Cells

SUBROUTINES: Common: ELMRED, ELEM1, ELEM2, ETYPA, ETYPB, ARCTAN, ARCCOS

DESCRIPTION:

The subroutine will unpack cards 1 and 2; if an error occurs a comment will be printed and the normal exit will be taken with the A Reg = 0. Having unpacked the cards, the subroutine checks the element type to see that it is either A or B; if not, the above error procedure is followed. The input elements are converted to N M elements and stored in VEBLOC. Corresponding classical elements are stored in DEBLOC; these will be used later to create a true delta element block (DEBLOC). The maximum number of sets of elements in VEBLOC and DEBLOC is 100.

1NXTCON
(SYS)
11/30/62

PURPOSE: To retrieve next control line

CALL SEQUENCE: JMP 1NXTCON

INPUTS: Magnetic tape, paper tape, or flexowriter
CONBITS

OUTPUTS: See Description

STORAGE REQUIREMENTS: Part of SYS

SUBROUTINES: None

DESCRIPTION:

This subroutine puts the next control line from the flexowriter, paper tape, or magnetic tape into core locations CONLINE through CONLINE + 9.

(1) Magnetic tape - Blank cards are ignored
(2) Flexowriter - First parameter CONLINE + 1 (Left Justified)
Second parameter CONLINE + 2 (Left Justified)
Etc.
(3) Paper tape - Same as Flexowriter

1NXTCON uses bits 1-15, 21-23, 42-47 of "CONBITS", therefore the user program must insure that "CONBITS" is properly set for its purposes.

1TYPOUT
(SYS)
11/30/62

PURPOSE: To transmit information to the flexowriter.

CALL SEQUENCE:

```
TMA    (WORD)
JMP    1TYPOUT
```

INPUTS: WORD = C/HLT, (Beginning address of comment); C/HLT, (No. of words)

OUTPUTS: Flexowriter output

STORAGE REQUIREMENTS: Part of SYS

SUBROUTINES: None

DESCRIPTION:

This subroutine will transmit N words to the flexowriter; words of trailing blanks are suppressed. However, it will not issue carriage returns, tabs, upper and lower case shifts.

SIMULATION ERROR ANALYSIS SYSTEM — PROGRAM DOCUMENT
SFXINT
(SEA)
11/30/62

PURPOSE: To convert BCD numbers to fixed point binary integers.

CALL SEQUENCE:

```
TMA (WORD 1)
TMQ (WORD 2)
JMP SFXINT
JAZ (ERROR)
TQM (RESULT)
```

INPUTS: WORD 1, WORD 2 } 16 character BCD number

OUTPUTS: RESULT - fixed point binary integer (T 47)

STORAGE REQUIREMENTS: 110 Cells

SUBROUTINES: None

DESCRIPTION:

All blanks are ignored. Only leading zeros are ignored; trailing zeros are significant. The conversion will terminate upon completion of converting 16 BCD characters, or finding a decimal point. If a decimal point is found, the result in the Q register will be the integral part of the number entered. The accumulator will not be zero in this case (i.e. no error).

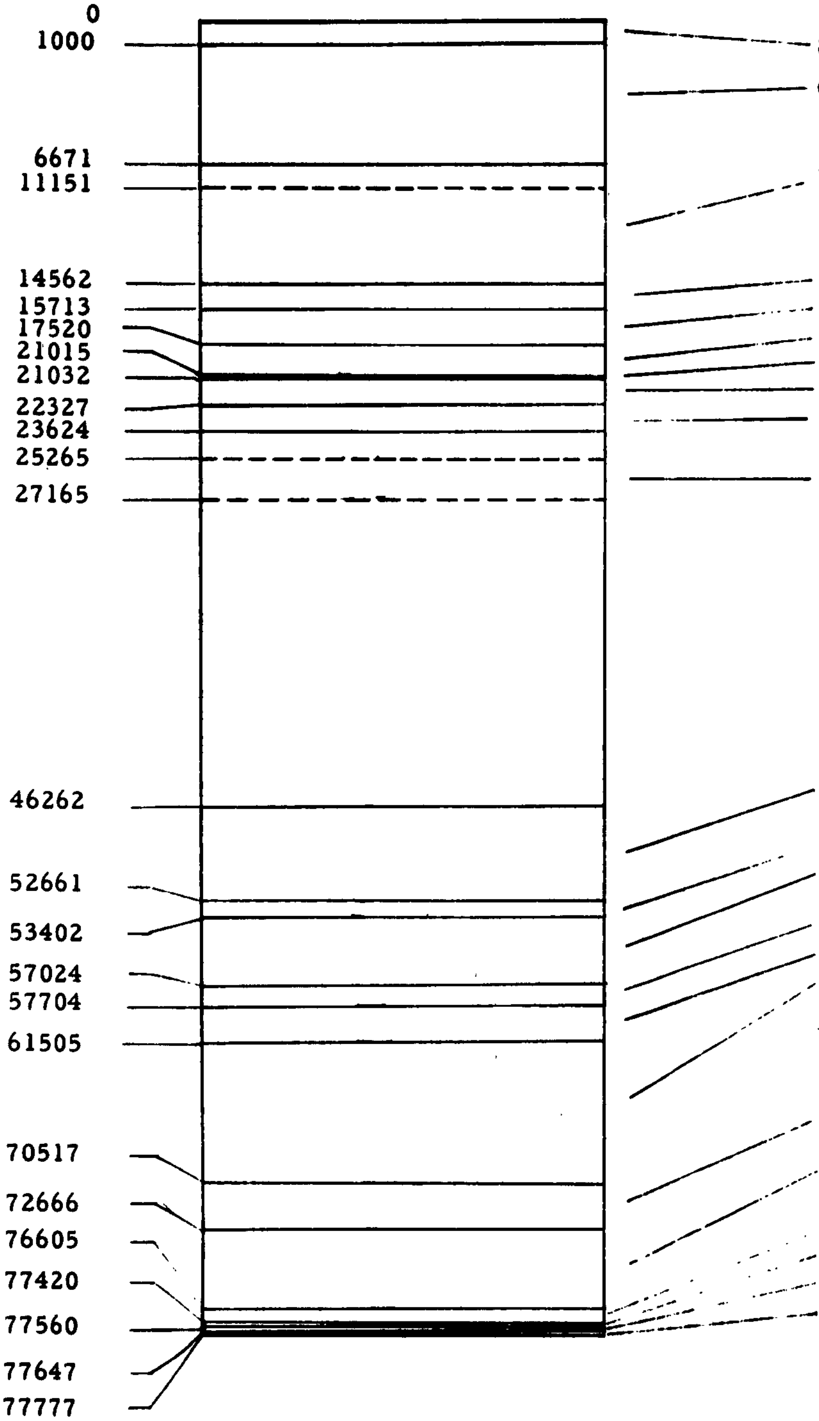
SECTION 6

SYSTEM BLOCK FORMATS

The data read into the Simulation Error Analysis System and much of the information subsequently generated must be readily accessible to the various parts of the system. For this reason, several blocks of data are defined in high speed core storage. These blocks have been assigned specific formats which are described herein, each under its particular block designation. The formats specify the words as floating point numbers and BCD characters. In general, the BCD characters are six bit characters yielding eight per word. Blank characters are designated by the symbol △.

In several instances, the blocks have an associated retrieval routine (see Subroutine Section for details), which will pick a desired entry from the block and place the data in assigned core locations. These assigned locations, which are not necessarily consecutive, and the corresponding retrieval routines are included where applicable. There are a few blocks which have no retrieval routines as such, but which have assigned storage locations where the associated quantities may be placed by the user. Other blocks have assigned storage locations with the block itself, and may be referred to directly by these assignments.

Figure 6-1 displays the storage assignments for all of the core block reservations. Storage assignments are also shown for the other system components.

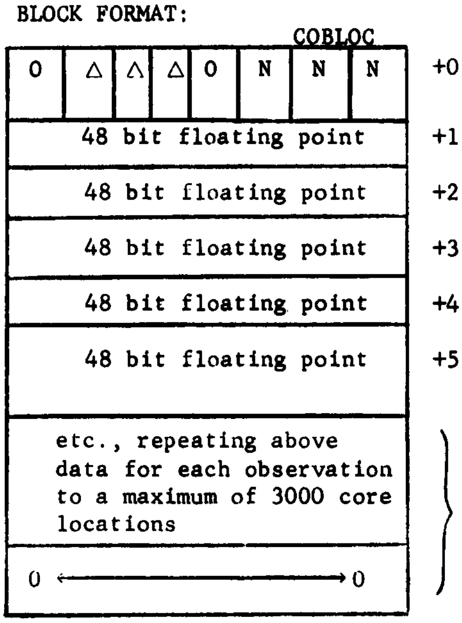


Use	Octal Size	Decimal Equiv.
SYS	1000_8	512_{10}
OBLOC	5671	3001
COBLOC	5671	3001
TBLOC	2261	1201
SBLOC	1131	601
SGBLOC	1605	901
IBLOC	1275	701
EBLOC	15	13
VEBLOC	1275	701
DEBLOC	1275	701
POBLOC	22434	9500
STPBLOC	1700	960
DPVBLOC	1441	801
Not used	4401	2305
Philco Subs	521	337
Exec	3422	1810
OBSIM	660	432
AQTINT	1601	897
Common Sub.	7012	3594
ERPRED	2147	1127
DC	3717	1999
MOCARO	613	395
INTERF1	140	96
CONTROL1	67	55
DCDIST	131	89

FIGURE 6-1

CORE ALLOCATION

BLOCK FORMAT:



COBLOC

O	Δ	Δ	Δ	O	N	N	N	+0
48 bit floating point								+1
48 bit floating point								+2
48 bit floating point								+3
48 bit floating point								+4
48 bit floating point								+5
etc., repeating above data for each observation to a maximum of 3000 core locations								
0 ⟷ 0								

O = Control Character, O = Observation Typre, N = Sensor Number

ρ = Range (Earth radii)

A, α = Azimuth or Right Ascension (Rad)

h, δ = Elevation or Declination (Rad)

$\dot{\rho}$ = Range rate (Earth radii/k_e^{-1} min)

t = Time since epoch (Minutes)

Data block is variable in length, but must be ended with a word of zeros

BLOCK GENERATION: COBLOC is generated in the MOCARO unit. It also may be generated for sequences which read in observation cards. In these cases, COBLOC is generated by the OCCARD subroutine for system observation cards or by the COBCRD subroutine for SPADATS observation cards.

BLOCK RETRIEVAL: Entries in COBLOC can be retrieved sequentially or by sensor number by the COBGET subroutine. An entry to INITCOB is required in order to initialize COBGET for the first COBLOC entry. Each reference to COBGET will retrieve the next COBLOC observation entry if the sensor number is not specified, placing the data in the following core locations and word formats:

△	△	△	△	△	N	N	N	STAID
0	0	0	0	0	0	oox	xxx	OTYPE - Bits 44-47 Set = 0, 1 (See Note)
48 bit floating point								RANGE - Earth radii
48 bit floating point								ALPHA - Radians
48 bit floating point								DELTA - Radians
48 bit floating point								RODOT - Earth radii/k_e^{-1} min
48 bit floating point								T - Minutes

NOTE: OTYPE has bits 44-47 set to 1 if the corresponding quantities were simulated.

BLOCK FORMAT:

DEBLOC

48 bit floating point	+0	Δa	(Earth radii)
48 bit floating point	+1	Δe	
48 bit floating point	+2	Δi	(Radians)
48 bit floating point	+3	$\Delta \Omega$	(Radians)
48 bit floating point	+4	$\Delta \omega$	(Radians)
48 bit floating point	+5	ΔM_o	(Radians)
48 bit floating point	+6	ΔC_o	(Days/Rev^2)
etc., repeating above data to a maximum of 700 core locations (100 sets)	}	Data block is variable in length but must be ended with a word of zeros	
0 ←——→ 0			

BLOCK GENERATION: DEBLOC is generated internally by the SGPDC unit. In SEA8 it is generated by the PREERP program which controls the reading of Element and Variant Element cards. In both cases the values are the result of variant elements minus nominal elements.

BLOCK RETRIEVAL: Entries in DEBLOC are retrieved sequentially by the DELGET subroutine. An entry to INITDEL is necessary to initialize DELGET to begin with the first set of delta elements. Each entry to DELGET will retrieve the next set of data and place it in the following core locations and word formats:

48 bit floating point	DELA	Δa (Earth radii)
48 bit floating point	DELE	Δe
48 bit floating point	DELI	Δi (Radians)
48 bit floating point	DELNOD	$\Delta \Omega$ (Radians)
48 bit floating point	DELOM	$\Delta \omega$ (Radians)
48 bit floating point	DELMO	ΔM_o (Radians)
48 bit floating point	DELCO	ΔC_o (Days/Rev^2)

BLOCK FORMAT:

DPVBLOC

DPVBLOC					
48 bit floating point	+0	Δx	or	D_U	(Earth radii)
48 bit floating point	+1	Δy		D_V	(Earth radii)
48 bit floating point	+2	Δz		D_W	(Earth radii)
48 bit floating point	+3	$\|\Delta \underline{r}\|$		$\|\Delta \underline{r}\|$	(Earth radii)
48 bit floating point	+4	$\Delta \dot{x}$		$\dot{D}_U$	(Earth radii/k_e^{-1} min)
48 bit floating point	+5	$\Delta \dot{y}$		$\dot{D}_V$	(Earth radii/k_e^{-1} min)
48 bit floating point	+6	$\Delta \dot{z}$		$\dot{D}_W$	(Earth radii/k_e^{-1} min)
48 bit floating point	+7	$\|\Delta \dot{\underline{r}}\|$		$\|\Delta \dot{\underline{r}}\|$	(Earth radii/k_e^{-1} min)
etc., repeating above data to a maximum of 800 words (100 sets)		Data block is variable in length, but must be ended with a word of zeros.			
0 ←——→ 0					

BLOCK GENERATION: DPVBLOC is generated by the ERP program unit. The contents of the block may be determined by testing core location OTPFLAG: OTPFLAG $\left\{\begin{matrix} =0 & xyz \\ =1 & UVW \end{matrix}\right\}$.

BLOCK RETRIEVAL: No retrieval subroutine exists for this block.

BLOCK FORMAT:

EBLOC			

E T A A A N N N	+0	E = Control Character, T = Element Type
		A = Element Number, N = Satellite Number	

48 bit floating point	+1	a_{xNo}	
48 bit floating point	+2	a_{yNo}	
48 bit floating point	+3	h_{xo}	$\frac{(\text{Earth radii})^2}{k_e^{-1}\text{min}}$
48 bit floating point	+4	h_{yo}	$\frac{(\text{Earth radii})^2}{k_e^{-1}\text{min}}$
48 bit floating point	+5	h_{zo}	$\frac{(\text{Earth radii})^2}{k_e^{-1}\text{min}}$
48 bit floating point	+6	L_o	(Radians)
48 bit floating point	+7	C_o	(Days/Rev^2)
48 bit floating point	+8	t_o = Time from beginning of year to epoch (Days)	
48 bit floating point	+9	θ_{to} = Greenwich Sidereal time at epoch (Radians)	
B C D	+10	YYMMDDHH	Epoch (blank if element type C, D)
B C D	+11	M M S S · SSS	
0 ⟵ ⟶ 0	+12		

Data block is fixed length (13 words)

BLOCK GENERATION: EBLOC is generated in the EXEC unit by the ELMRED subroutine. Element cards 1 and 2 are interpreted and converted to $\underline{N}$ $\underline{M}$ elements, t_o and θ_{to} are computed, and EPOCH time is transferred from Element Card 2 to this block. Only one set of elements is allowable for each job.

BLOCK RETRIEVAL: Entries in EBLOC are retrieved by the EUNPCK subroutine. Since there is one set of elements, initialization of the routine is unnecessary. This routine places the data in the following core locations:

Δ	Δ	Δ	Δ	Δ	N	N	N	SATN	Satellite Number
0	0	0	0	0	A	A	A	ELNO	Element Number
0	0	0	0	0	0	0	T	ELMTYP	Element Type
0	0	0	0	0	0	0	Y	REFYEAR	Year of epoch
48 bit floating point								AXNO	
48 bit floating point								AYNO	
48 bit floating point								HXO	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point								HYO	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point								HZO	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point								XLO	(Radians)
48 bit floating point								CO	(Days/Rev^2)
48 bit floating point								EPOCH	Same as TO (Days)
Y	Y	M	M	D	D	H	H	YY	
M	M	S	S	·	S	S	S	MM	
48 bit floating point								THGRTO	θ_{to} (Radians)

BLOCK FORMAT:

IBLOC

I △ △ V O N N N	+0	I = Control Character, V = V Type, O = O Type, N = Sensor Number	
48 bit floating point	+1	h_i = Initial elevation (Radians)	+ = AM
48 bit floating point	+2	h_f = Final elevation (Radians)	- = PM
48 bit floating point	+3	ρ_{max} = Maximum range (Earth radii)	
48 bit floating point	+4	Δt = Observation interval (Minutes)	
0 0 0 0 X X X X X X X X	+5	X = Segments to use (4 Bit BCD)	
0 0 0 0 0 0 0 N	+6	N = Number of segments	
etc., repeating data for each sensor to a maximum of 700 core locations		Data block is variable in length but must be ended with a word of zeros.	
0 ← → 0			

BLOCK GENERATION: IBLOC is generated in the EXEC unit by subroutine ICARD. All of the quantities are input on Simulation Parameter cards (I cards).

BLOCK RETRIEVAL: Entries in IBLOC are retrieved by sensor number by the subroutine IGET. An entry to INITI is required once to initialize counters and addresses. Each reference to IGET will retrieve the data corresponding to the specified sensor and place it in the following core locations and word formats:

△ △ △ △ △ N N N	STAID	- Sensor Number
0 0 0 0 0 0 0 0	OTYPE	- (See OBLOC format)
0 0 0 0 0 0 0 V	VTYPE	- V = Visual, △ = Non-Visual
0 0 0 0 0 0 X X	DIRI	X X = 00 (AM) 11 (PM)
0 0 0 0 0 0 X X	DIRF	
0 0 0 0 0 0 0 N	NUSEG	- Number of segments
0 0 0 0 X X X X X X X X	SEGNOS	- Segments to use (4 Bit BCD)
48 bit floating point	HI	- Initial elevation (Radians)
48 bit floating point	HF	- Final elevation (Radians)
48 bit floating point	IDELTAT	- Observation frequency (Minutes)
48 bit floating point	ROMAX	- Maximum range (Earth radii)

BLOCK FORMAT:

OBLOC		
O △ △ △ 0 N N N	+0	O = Control Character, 0 = Observation Type, N = Sensor Number
48 bit floating point	+1	ρ = Range (Earth radii)
48 bit floating point	+2	A, α = Azimuth or Right Ascension (Rad)
48 bit floating point	+3	h, δ = Elevation or Declination (Rad)
48 bit floating point	+4	$\dot{\rho}$ = Range rate (Earth radii/k_e^{-1} min)
48 bit floating point	+5	t = Time since epoch (Minutes)
etc., repeating above data for each observation to a maximum of 3000 core locations		Data block is variable in length, but must be ended with a word of zeros.
$0 \longleftrightarrow 0$		

BLOCK GENERATION: OBLOC is generated in the OBSIM unit. It also may be generated for sequences which read in observation cards. In these cases, OBLOC is generated by the OBCARD subroutine for system observation cards or by the OBSCARD subroutine for SPADATS observations cards.

BLOCK RETRIEVAL: Entries in OBLOC are retrieved sequentially by the OBSGET subroutine. An entry to INITOBS is required in order to initialize OBSGET for the first OBLOC entry. Each reference to OBSGET will retrieve the next OBLOC observation entry, placing the data in the following core locations and word formats:

<table>
<tr><td>△</td><td>△</td><td>△</td><td>△</td><td>△</td><td>N</td><td>N</td><td>N</td><td>STAID</td><td></td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>oox</td><td>xxx</td><td>OTYPE</td><td>-</td><td>Bits 44-47 Set = 0, 1 (See Note)</td></tr>
<tr><td colspan="8">48 bit floating point</td><td>RANGE</td><td>-</td><td>Earth radii</td></tr>
<tr><td colspan="8">48 bit floating point</td><td>ALPHA</td><td>-</td><td>Radians</td></tr>
<tr><td colspan="8">48 bit floating point</td><td>DELTA</td><td>-</td><td>Radians</td></tr>
<tr><td colspan="8">48 bit floating point</td><td>RODOT</td><td>-</td><td>Earth radii/k_e^{-1} min</td></tr>
<tr><td colspan="8">48 bit floating point</td><td>T</td><td>-</td><td>Minutes</td></tr>
</table>

NOTE: OTYPE has bits 44-47 set to 1 if the corresponding quantities were simulated

BLOCK FORMAT:

SBLOC

S △ △ △ △ N N N	+0	S = Control Character,
		N = Sensor Number

48 bit floating point	+1	λ = Longitude, + East (Rad)
48 bit floating point	+2	ϕ = Latitude (Rad)
48 bit floating point	+3	h = Altitude (Earth radii)
48 bit floating point	+4	$X/\text{Cos}\ \theta = -(C + H) \cos \Phi$ (Earth radii)
48 bit floating point	+5	$Z = -(S + H) \sin \Phi$ (Earth radii)
etc., repeating above data for each sensor to a maximum of 600 core locations		Data block is variable in length but must be ended with a word of zeros.
0 ⟷ 0		

BLOCK GENERATION: SBLOC is generated in the EXEC unit by the SENCARD subroutine. All the quantities are input on Sensor cards (S cards) except for Z and $X/\cos\theta$ which are computed.

BLOCK RETRIEVAL: Entries in SBLOC can be retrieved sequentially or by sensor number by the SENGET subroutine. An entry to INITSEN is required to initialize SENGET to begin with the first sensor. The sensor number will be placed in STAID only if sequential retrieval is used. Each reference to SENGET will retrieve the data for the next sensor or the specified sensor and place it in the following core locations and word formats:

48 bit floating point	XLAMBA	(Radians)
48 bit floating point	PHIRD	(Radians)
48 bit floating point	OALT	(Earth radii)
48 bit floating point	XOVCT	(Earth radii)
48 bit floating point	CAPZ	(Earth radii)
Λ Λ Λ Λ Λ N N N	STAID	Sensor Number

BLOCK FORMAT:

SGBLOC

G △ △ △ △ N N N	+0	G = Control Character,
		N = Sensor Number	

48 bit floating point	+1	σ_ρ	(Earth radii)
48 bit floating point	+2	σ_α (or) σ_A	(Radians)
48 bit floating point	+3	σ_δ (or) σ_h	(Radians)
48 bit floating point	+4	$\sigma_{\dot\rho}$	(Earth radii/k_e^{-1} min)
48 bit floating point	+5	$\Delta\rho$	(Earth radii)
48 bit floating point	+6	Δ_α (or) σA	(Radians)
48 bit floating point	+7	$\Delta\delta$ (or) Δh	(Radians)
48 bit floating point	+8	$\Delta\dot\rho$	(Earth radii/k_e^{-1} min)
etc., repeating above data for each sensor to a maximum of 900 core locations		Data block is variable in length, but must be ended with a word of zeros.	
0 ⟷ 0			

BLOCK GENERATION: SGBLOC is generated in the EXEC unit by the MCARD subroutine. The quantities are input values which are entered on Standard Deviations cards (M cards) and are not computed by any program unit.

BLOCK RETRIEVAL: Entries in SGBLOC are retrieved by sensor number by the SIGGET subroutine. An entry to INITSIG is necessary to initialize SIGGET. Each entry to SIGGET will retrieve the set of sigmas and biases corresponding to the sensor specified and place the data in the following core locations and word formats:

Δ Δ Δ Δ Δ N N N	STAID	Sensor Number
48 bit floating point	SIGRHO	(Earth radii)
48 bit floating point	SIGALP	(Radians)
48 bit floating point	SIGDLT	(Radians)
48 bit floating point	SIGRRT	(Earth radii/k_e^{-1} min)
48 bit floating point	BIASRHO	(Earth radii)
48 bit floating point	BIASALP	(Radians)
48 bit floating point	BIASDLT	(Radians)
48 bit floating point	BIASRRT	(Earth radii/k_e^{-1} min)

BLOCK FORMAT: STPBLOC

48 bit floating point	+0	μ	(Δx)	or	μ (D_U)	(Earth radii)
48 bit floating point	+1	σ	(Δx)		σ (D_U)	(Earth radii)
48 bit floating point	+2	Max	(Δx)		Max (D_U)	(Earth radii)
48 bit floating point	+3	μ	(Δy)		μ (D_V)	(Earth radii)
48 bit floating point	+4	σ	(Δy)		σ (D_V)	(Earth radii)
48 bit floating point	+5	Max	(Δy)		Max (D_V)	(Earth radii)
48 bit floating point	+6	μ	(Δz)		μ (D_W)	(Earth radii)
48 bit floating point	+7	σ	(Δz)		σ (D_W)	(Earth radii)
48 bit floating point	+8	Max	(Δz)		Max (D_W)	(Earth radii)
48 bit floating point	+9	μ	$\lvert\Delta \underline{r}\rvert$		μ $\lvert\Delta \underline{r}\rvert$	(Earth radii)
48 bit floating point	+10	σ	$\lvert\Delta \underline{r}\rvert$		σ $\lvert\Delta \underline{r}\rvert$	(Earth radii)
48 bit floating point	+11	Max	$\lvert\Delta \underline{r}\rvert$		Max $\lvert\Delta \underline{r}\rvert$	(Earth radii)
48 bit floating point	+12	μ	$(\Delta\dot{x})$		μ $(\dot{D}_U)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+13	σ	$(\Delta\dot{x})$		σ $(\dot{D}_U)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+14	Max	$(\Delta\dot{x})$		Max $(\dot{D}_U)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+15	μ	$(\Delta\dot{y})$		μ $(\dot{D}_V)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+16	σ	$(\Delta\dot{y})$		σ $(\dot{D}_V)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+17	Max	$(\Delta\dot{y})$		Max $(\dot{D}_V)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+18	μ	$(\Delta\dot{z})$		μ $(\dot{D}_W)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+19	σ	$(\Delta\dot{z})$		σ $(\dot{D}_W)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+20	Max	$(\Delta\dot{z})$		Max $(\dot{D}_W)$	(Earth radii/k_e^{-1} min)
48 bit floating point	+21	μ	$\lvert\Delta \dot{\underline{r}}\rvert$		μ $\lvert\Delta \dot{\underline{r}}\rvert$	(Earth radii/k_e^{-1} min)
48 bit floating point	+22	σ	$\lvert\Delta \dot{\underline{r}}\rvert$		σ $\lvert\Delta \dot{\underline{r}}\rvert$	(Earth radii/k_e^{-1} min)
48 bit floating point	+23	Max	$\lvert\Delta \dot{\underline{r}}\rvert$		Max $\lvert\Delta \dot{\underline{r}}\rvert$	(Earth radii/k_e^{-1} min)
etc., repeating above data up to a maximum of 960 core locations	}	Data block is variable length but must be ended with a word of zeros.				
0 ←→ 0						

BLOCK GENERATION: STPBLOC is generated by the ERPRED unit. The contents of the block may be determined by testing core location OTPFLAG.

$$\text{OTPFLAG} = \left\{ \begin{array}{l} = 0 \rightarrow \text{xyz} \\ = 1 \rightarrow \text{UVW} \end{array} \right\} .$$

BLOCK RETRIEVAL: No retrieval routine exists for this block.

BLOCK FORMAT:

TBLOC

0 Δ Δ V O N N N	+0	O = Control Character, V = VTYPE, O = OTYPE, N = Sensor Number	
48 bit floating point	+1	Δ_t (Minutes)	
48 bit floating point	+2	ρ_{max} = Maximum Range (Earth radii)	
48 bit floating point	+3	t_i = Beginning Time (Minutes)	First Pair
48 bit floating point	+4	t_f = Ending Time (Minutes)	
↓			
48 bit floating point		t_i = Beginning Time (Minutes)	nth Pair
48 bit floating point		t_f = Ending Time (Minutes)	
etc., listing all acquisition times sensor by sensor to a maximum of 1200 words		Data block is variable in length and may contain acquisition data for many sensors. The block must be ended by a word of zeros.	
0 ←——→ 0			

BLOCK GENERATION: TBLOC is generated by the AQTINT unit, being the primary output of that program.

BLOCK RETRIEVAL: OBSIM uses TBLOC in the above format. There is no specific retrieval routine for TBLOC.

BLOCK FORMAT:

VEBLOC			
48 bit floating point	+0	a_{xNo}	
48 bit floating point	+1	a_{yNo}	
48 bit floating point	+2	h_{xo}	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point	+3	h_{yo}	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point	+4	h_{zo}	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point	+5	L_o	(Radians)
48 bit floating point	+6	C_o	(Days/Rev^2)
etc., repeating above data to a maximum of 700 core locations (100 sets)	}	Data block is variable in length but must be ended with a word of zeros.	
0 ←——→ 0			

BLOCK GENERATION: VEBLOC is generated internally by the SGPDC unit. It may also be generated for SEA8 by the EXEC unit subroutine VECARD, which reads Variant Element cards (V cards) into this block.

BLOCK RETRIEVAL: Entries in VEBLOC are retrieved sequentially by the ELMGET subroutine. An entry to INITELM is necessary to initialize ELMGET to start with the first set of variant elements. Each entry to ELMGET will retrieve the next set of data and place it in the following core locations and word formats:

48 bit floating point	AXNO	
48 bit floating point	AYNO	
48 bit floating point	HXO	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point	HYO	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point	HZO	$\frac{(\text{Earth radii})^2}{k_e^{-1}\ \text{min}}$
48 bit floating point	XLO	(Radians)
48 bit floating point	CO	(Days/Rev^2)

SECTION 7

SYSTEM CARD FORMATS

The card formats compatible with the Simulation Error Analysis System fall into three main categories: SYS cards, SEA cards, and Data cards. SYS cards are designed to be used by SYS and are processed by SYS. SEA cards are designed for the SEA System and are processed by the SEA EXEC. The Data cards are designed to provide input data to the SEA System and are also processed by the SEA EXEC.

SYS Cards

The card function of SYS cards is given in Columns 17-24. The following SYS cards are allowed during an SEA run (reference should be made to Philco SYS documentation for interpretation and format details):

DUMP
SNAP
OCT
COMMAND
JMP
JMPL
JMPR

CONIN
HLT
REM
IBIT
ORIGIN
COMMON
MASTER
SEGMENT

REWIND
REWINDLO
LOCSENT
WRTSENT
READF
READB
WRITE
CLOCK

SEA Cards

The card function for SEA cards is given in Columns 17-24. The SEA cards defined to date are:

Name	Figure
JOB	7-1
FLOAT	7-2
PRINT	7-3
RPL	7-4
SEA	7-5
ENDDATA	7-20
SYS	7-21
CONTROL	
AQTINT	7-11
MOCARO	7-12
SGPDC	7-13
ERPRED	7-14

Data Cards

The function of Data card is indicated by a "card type" in Column 80. This card type is one BCD letter assigned to that card. Data cards defined to date include:

Name	Card Type	Figure
SENSOR	S	7-6
SIMULATION PARAMETER	I	7-7
STANDARD DEVIATION	M	7-8
ELEMENT CARD 1	E	7-9
ELEMENT CARD 2	E	7-10
SYSTEM OBSERVATION	O	7-15
SPADATS OBSERVATION	△,0-9	7-16
VARIANT ELEMENT 1	V	7-17
VARIANT ELEMENT 2	V	7-18
DISTRIBUTION	Q	7-19

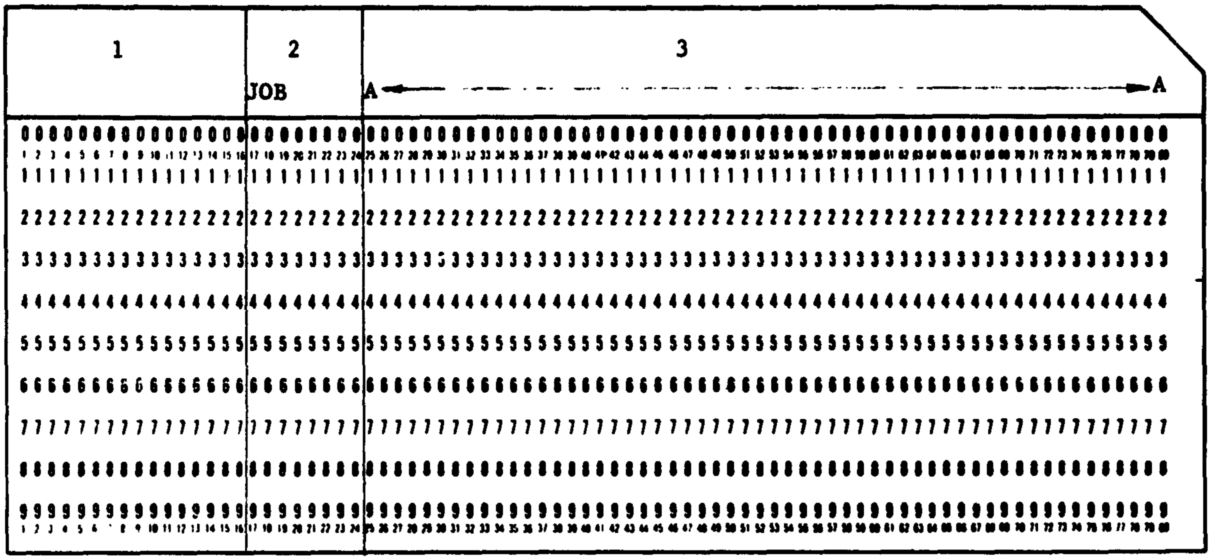
In the card figures, the letters appearing at the top of a column in the fields of the card image, have the following meanings:

A - Entry in this field consists of letters and in general form, a word for identification purposes

F - Entry in this field will be interpreted as a floating point number

I - Entry in this field will be interpreted as a fixed point integer

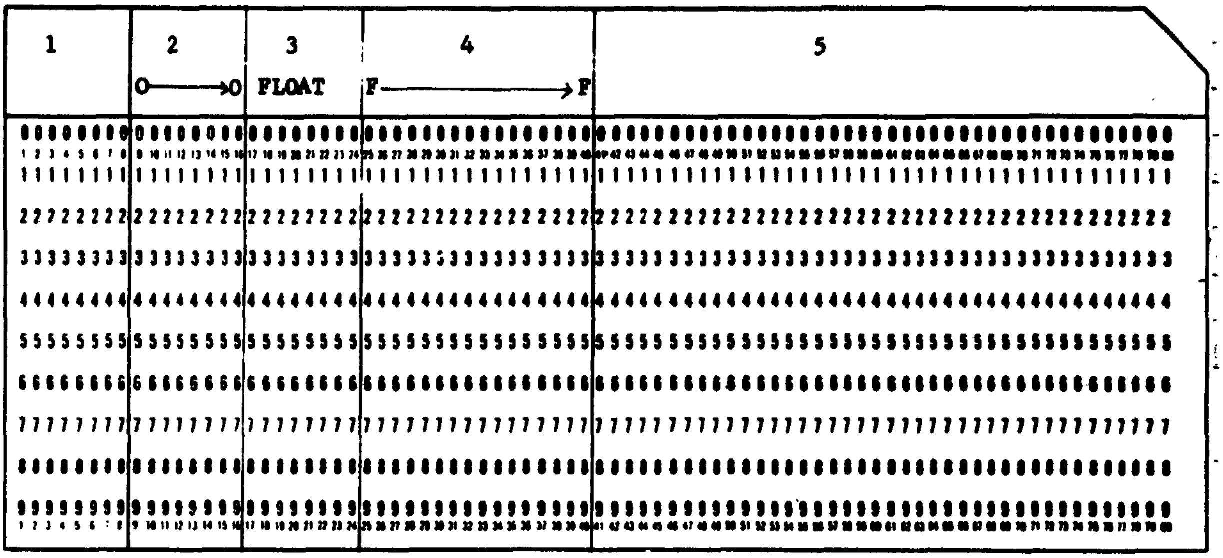
O - Entry in this field will be interpreted as octal format.



Field	Column	Description
1	1 - 16	Not used.
2	17 - 24	JOB (card function). Defines start of SYS Job.
3	25 - 80	JOB description for identification purposes. Appears at the beginning and end of output and on flexowriter.

FIGURE 7-1

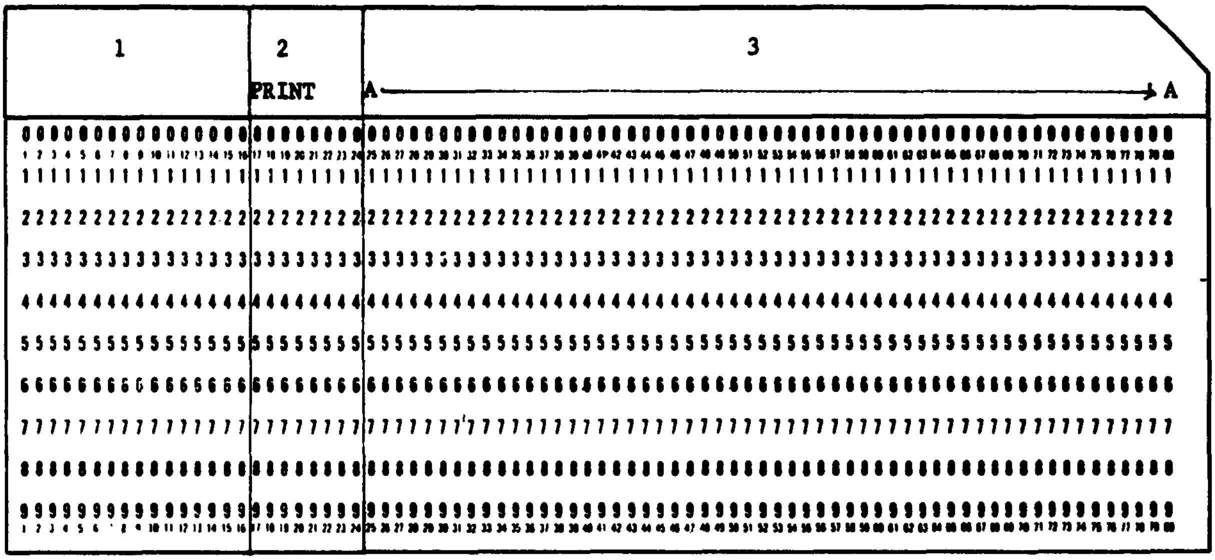
JOB CARD



Field	Column	Description
1	1 - 8	Not used.
2	9 - 16	Location to be corrected in octal.
3	17 - 24	FLOAT (card function) tells executive program to make floating point correction.
4	25 - 40	Quantity desired in the floating point format.
5	41 - 80	Not used.

FIGURE 7-2

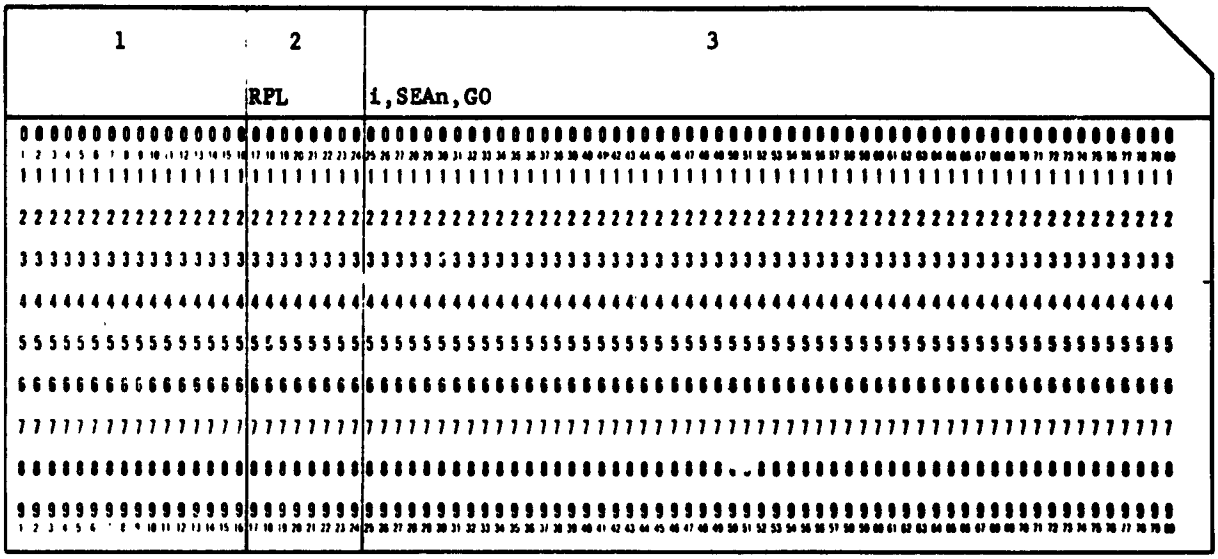
FLOAT CARD



Field	Column	Description
1	1 - 16	Not used.
2	17 - 24	PRINT (card function) signals EXEC to print following information on hard copy output.
3	25 - 80	BCD information to be printed.

FIGURE 7-3

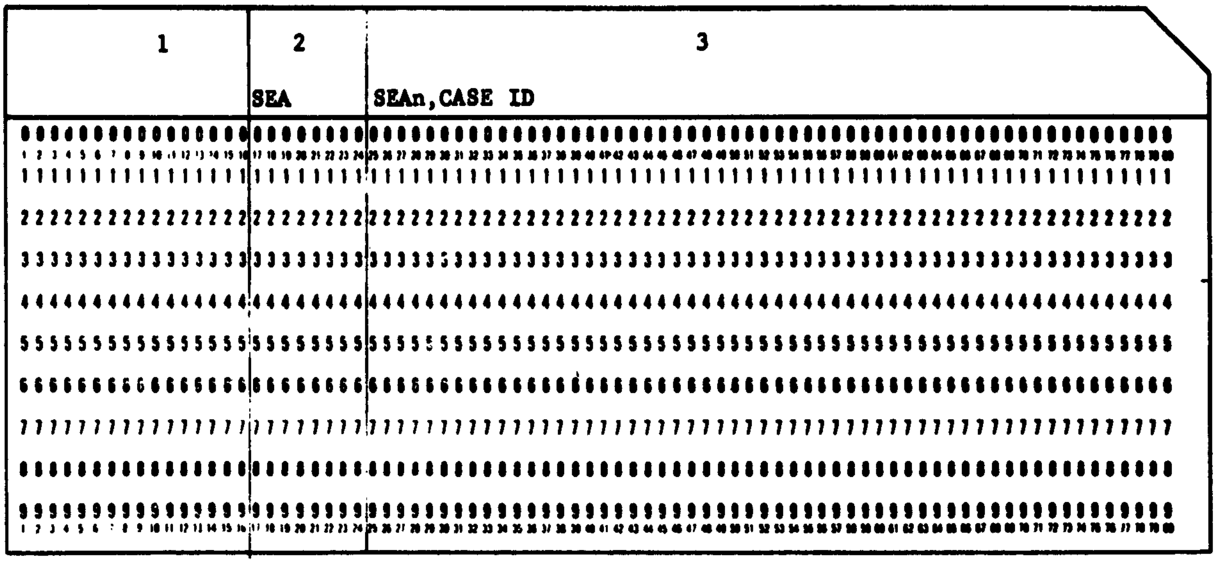
PRINT CARD



Field	Column	Description
1	1 - 16	Not used.
2	17 - 24	RPL (card function) calls RPL program into core from magnetic tape.
3	25 - 80	Variable field (parameters separated by any of the characters , . / ;) 1st parameter - Logical tape number where RPL program is located. Usually tape number 4. 2nd parameter - Program identity. Usually = SEAn where n = number of sequence to be run. (e.g., SEA1 or SEA2, etc.). 3rd parameter - "GO" - Indicates to sys that control is to be transferred to the RPL program after it is loaded.

FIGURE 7-4

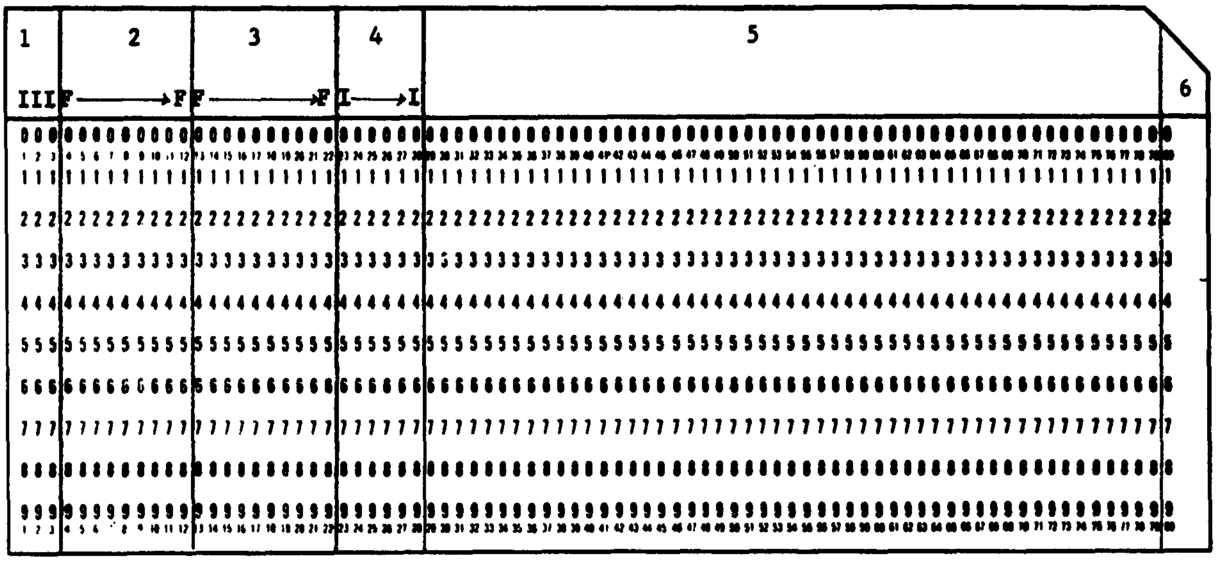
RPL CARD



Field	Column	Description
1	1 - 16	Not used.
2	17 - 24	SEA (card function) defines the beginning of a case.
3	25 - 80	Variable field, (parameters separated by any of the characters . , / ;) 1st parameter - code name of sequence being run (e.g. SEA1, SEA2, etc). 2nd parameter - 8 character (including blanks) case identification.

FIGURE 7-5

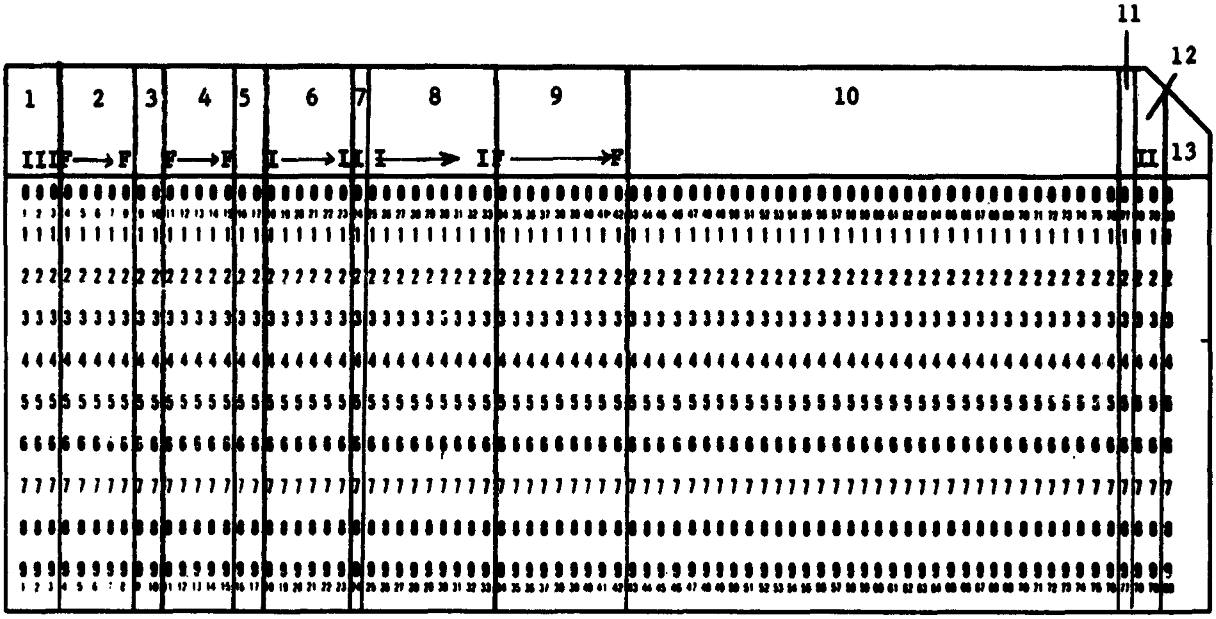
SEA CARD



Field	Column	Description
1	1 - 3	Sensor Number.
2	4 - 12	Ø-Latitude (degrees, + = North)
3	13 - 22	λ_E- Longitude (degrees, + = East)
4	23 - 28	H - Altitude (meters)
5	29 - 79	Not used
6	80	Card type = S

FIGURE 7-6

SENSOR CARD



Field	Column	Description
1	1 - 3	Sensor number
2	4 - 8	Beginning H for station (degrees)
3	9 - 10	AM or PM for Field 2
4	11 - 15	Final H for station (degrees)
5	16 - 17	AM or PM for Field 4
6*	18 - 23	Maximum range for station (km)
7	24	Number of segments to divide pass into: (integer 1-9, Δ= 1)
8	25 - 33	Segments to use: 1-9 (sequential), Δ= Entire pass
9*	34 - 42	Δt (observation time interval in minutes)
10	43 - 76	Not used
11	77	Visual = V, Non-visual =Δ
12	78 - 79	Observation type (see observation type table)
13	80	Card type = I

* Not used in Sequence 2

FIGURE 7-7

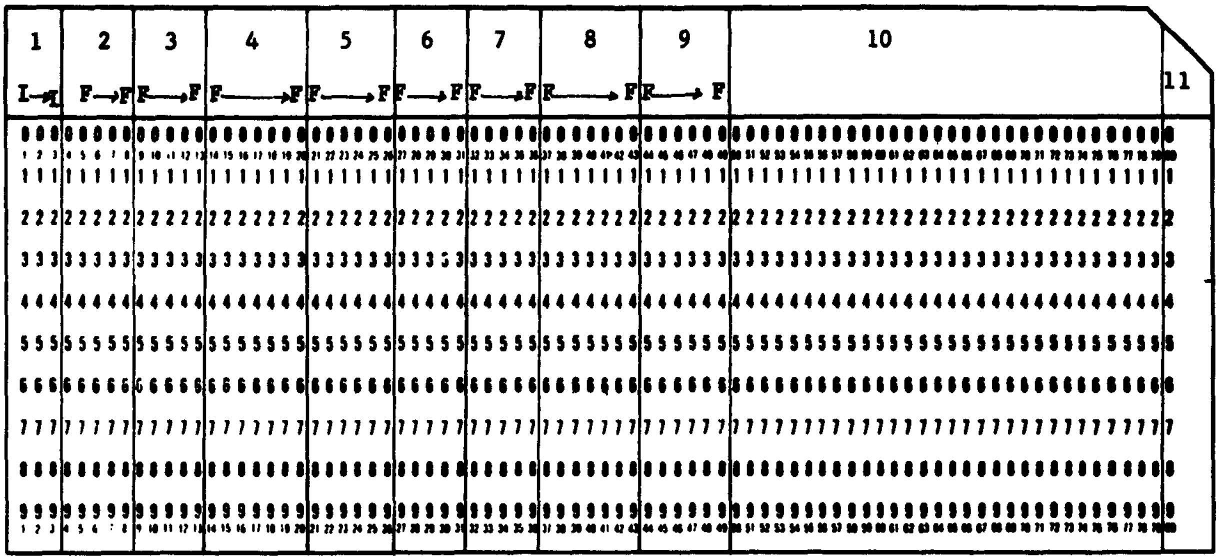
SIMULATION PARAMETER CARD

OBSERVATION TYPE TABLE

Observation Type	Time	Azimuth and Elevation	Rt. Ascen. and Declination	Range	Range Rate
1	X	X			
2	X	X		X	
3	X	X			X
4	X	X		X	X
5	X		X		
6	X		X	X	
7	X		X		X
8	X		X	X	X
9	X			X	
10	X			X	X
11	X				X

FIGURE 7-7 (cont.)

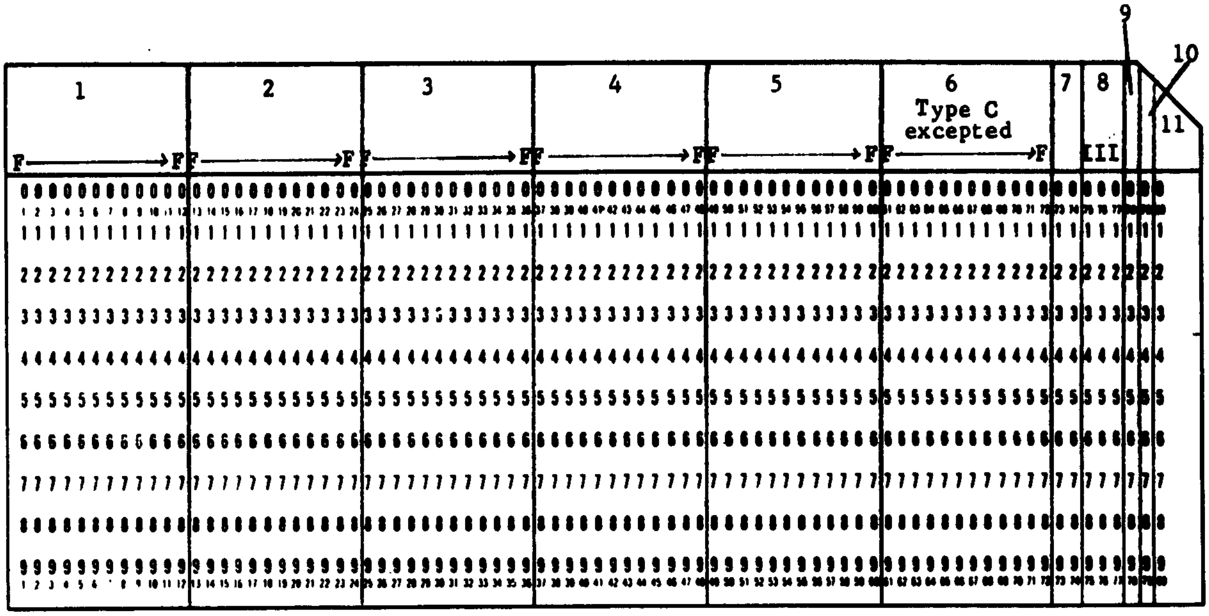
SIMULATION PARAMETER CARD



Field	Column	Description
1	1 - 3	Sensor number.
2	4 - 8	Standard deviation of azimuth or right ascension (degrees).
3	9 - 13	Standard deviation of elevation or declination (degrees).
4	14 - 20	Standard deviation of range (meters).
5	21 - 26	Standard deviation of range rate (meters per second).
6	27 - 31	Bias on azimuth or right ascension (degrees).
7	32 - 36	Bias on elevation or declination (degrees).
8	37 - 43	Bias on range (meters).
9	44 - 49	Bias on range rate (meters per second).
10	50 - 79	Not used.
11	80	Card type = M.

FIGURE 7-8

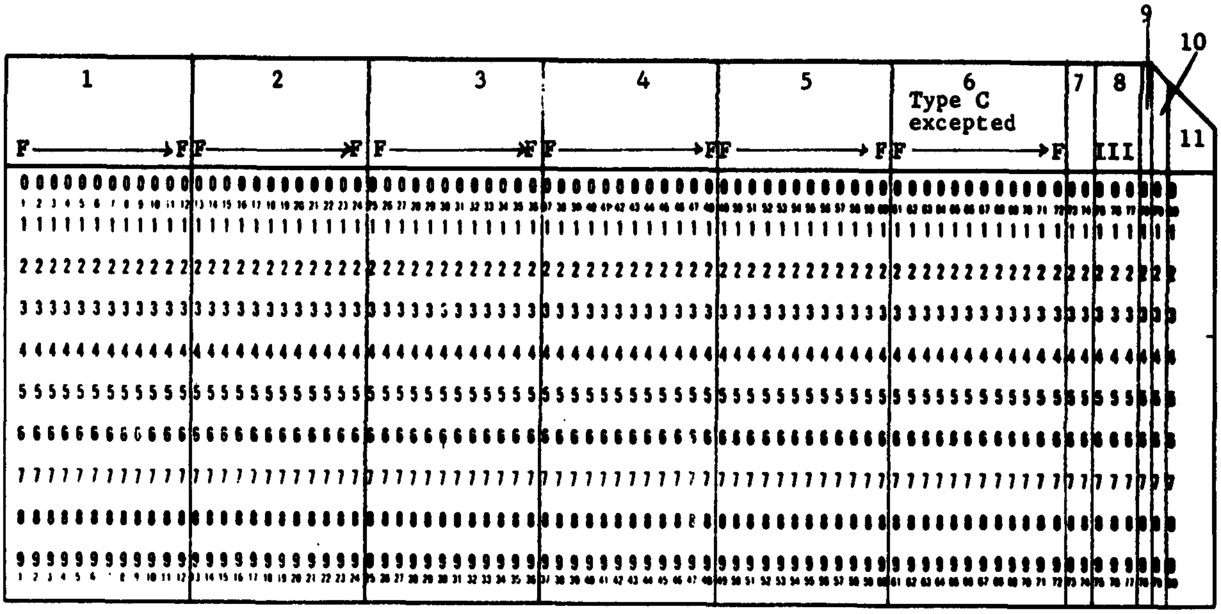
STANDARD DEVIATIONS CARD



Field	Column	Type A (Classical)	Type B (Position & Velocity)
1	1 - 12	a - Semi-major axis (earth radii)	x - Equator plane position component along ♈ (km)
2	13 - 24	e - Orbital eccentricity	y - Equator plane position component to x (km)
3	25 - 36	i - Orbital inclination (degrees)	z - Position component orthogonal to x, y (km)
4	37 - 48	Ω - Right ascension of ascending node (degrees)	$\dot{x}$ - Equator plane velocity component along ♈ (m/sec)
5	49 - 60	ω - Argument of perigee (degrees)	$\dot{y}$ - Equator plane velocity component to $\dot{x}$ (m/sec)
6	61 - 72	M_o - Mean anomaly at Epoch (degrees)	$\dot{z}$ - Velocity component orthogonal to $\dot{x}$, $\dot{y}$ (m/sec)
7	73 - 74	Not used	Not used
8	75 - 77	Satellite Number	Satellite Number
9	78	Card Number = 1	Card Number = 1
10	79	Element Type = A	Element Type = B
11	80	Card Type = E	Card Type = E

FIGURE 7-9

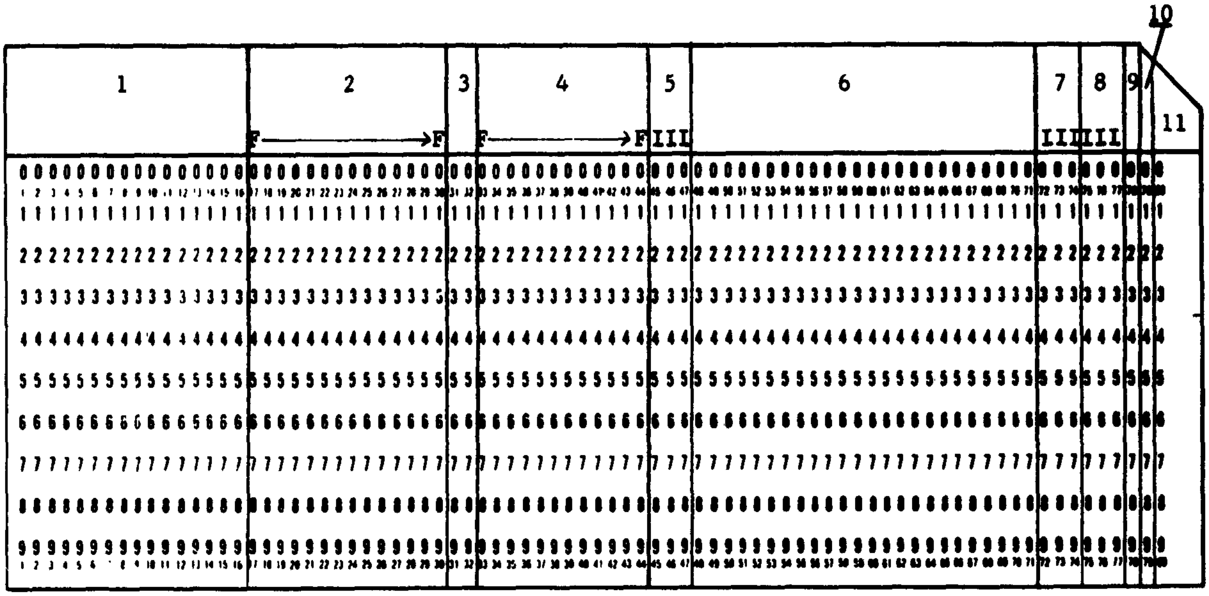
ELEMENT CARD 1



Field	Column	Type C (Orbit - Station)	Type D (Radar Oriented)
1	1 - 12	Hq - Altitude of perigee above reference ellipsoid (km)	a - Semi-major axis (earth radii)
2	13 - 24	e - Orbital eccentricity	e - Orbital eccentricity
3	25 - 36	i - Orbital inclination (degrees)	i - Orbital inclination (degrees)
4	37 - 48	h_c - Altitude angle at culmination over station (degrees)	Ω - Right ascension of ascending node (degrees)
5	49 - 60	v_c - True anomaly at culmination over station (degrees)	ω - Argument of perigee (degrees)
6	61 - 72	COL 65-69 NORTH SOUTH } North or South of Zenith COL 71-72 NS SN } Pass N-S or S-N over Stn.	h_o - Initial elevation angle (degrees)
7	73 - 74	Not Used	Not Used
8	75 - 77	Satellite Number	Satellite Number
9	78	Card Number = 1	Card Number = 1
10	79	Element Type = C	Element Type = D
11	80	Card Type = E	Card Type = E

FIGURE 7-9 Cont.

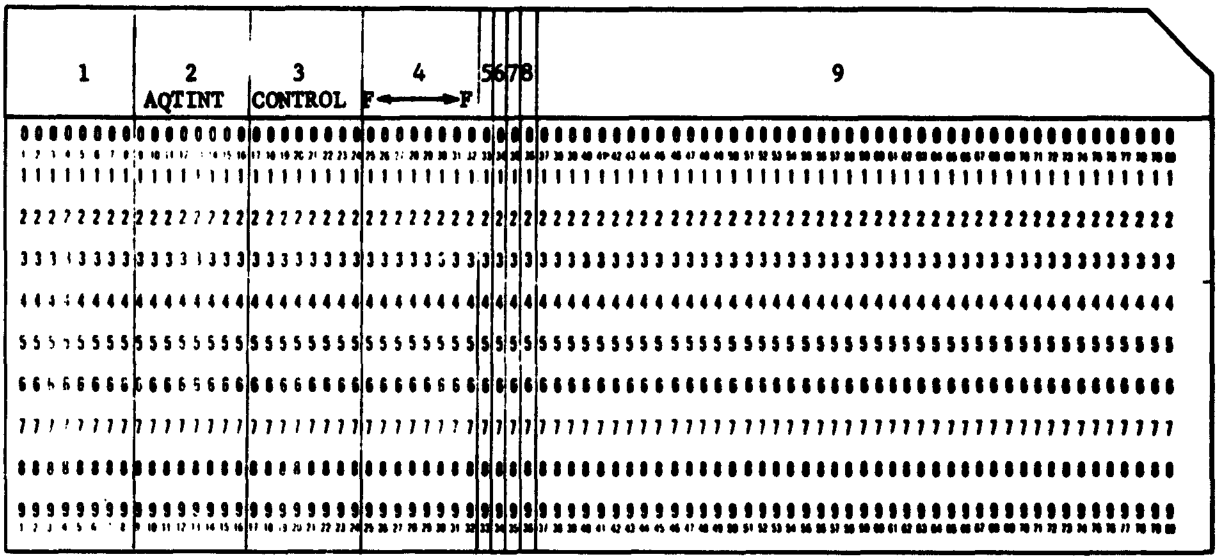
ELEMENT CARD 1



Field	Column	Description
1	1 - 16	Time of epoch (YYMMDDHHMMSS.SSS) Leave blank for types C & D
2	17 - 30	C - Rate of change of period (days/rev^2)-blank field = 0
3	31 - 32	Not used.
4	33 - 44	ρ_o - Initial range, element type D only - blank for types A, B, C
5	45 - 47	Sensor number (type C or D only)-blank for types A & B
6	48 - 71	Not used.
7	72 - 74	Element number
8	75 - 77	Satellite number
9	78	Card number = 2
10	79	Element type (A, B, C, D)
11	80	Card type = E

FIGURE 7-10

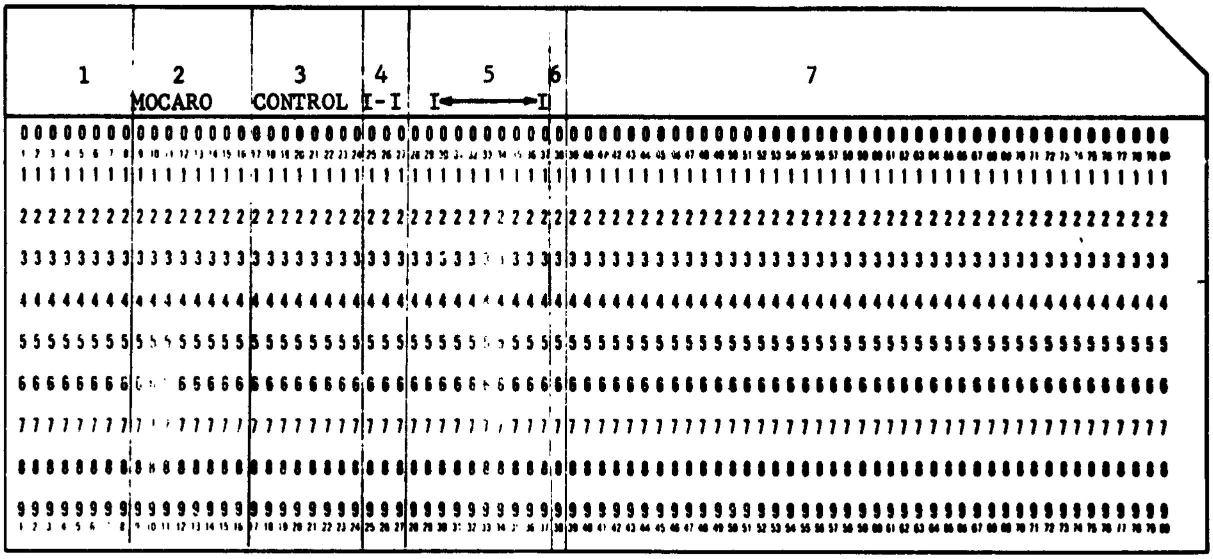
ELEMENT CARD 2



Field	Column	Description
1	1 - 8	Not used.
2	9 - 16	AQTINT (program ID)
3	17 - 24	CONTROL (card function)
4	25 - 32	Observation duration, minutes
5	33	Output option: Acquisition Times. 0 or ▲ , = No output; 1 = output
6 *	34	Output option: Observation Simulation. 0 or ▲ , no output; 1, output all Obs; 2, output 1st, middle, and last Obs. in each segment.
7	35	Perturbations control: 0 or ▲, No perturbations; 1, General perturbations
8 *	36	Punch option: Observation Simulation. 0 or △ , no punched cards; 1, punch all generated observations
9	37 - 80	Not used.

* Not used in Sequence 2

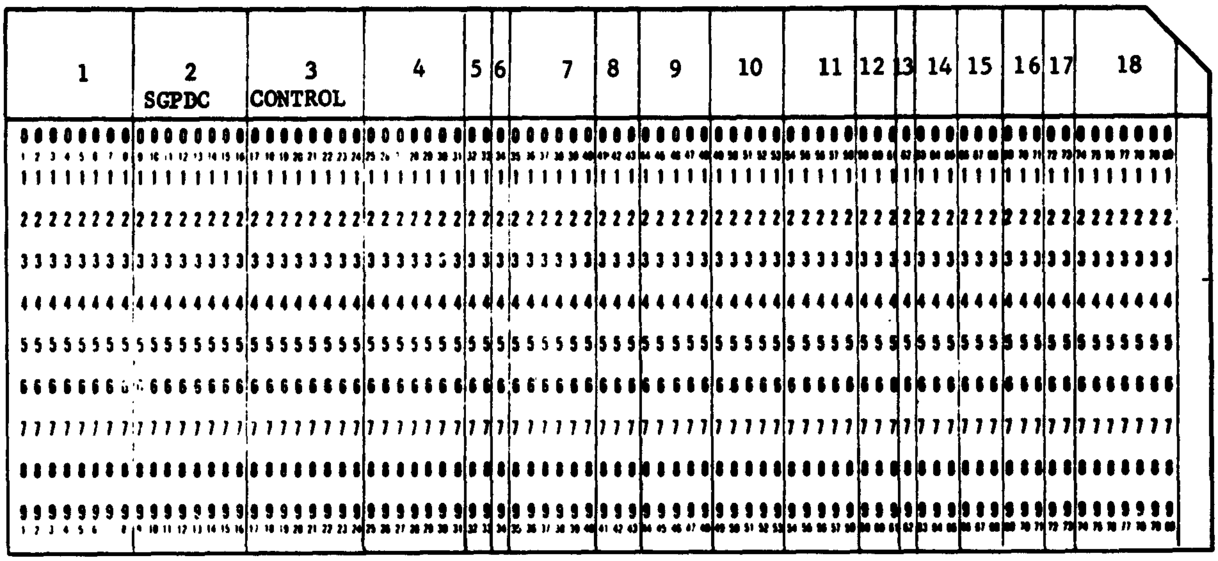
FIGURE 7-11
AQTINT CONTROL CARD



Field	Column	Description
1	1 - 8	Not used.
2	9 - 16	MOCARO (program ID)
3	17 - 24	CONTROL (card function)
4	25 - 27	K, number of contaminated sets of observations to be generated.
5	28 - 37	N, starting random number, integer of the form 4 I+1, justified right
6	38	Output option: 0 or △ , no output; 1, output.
7	39 - 80	Not used.

FIGURE 7-12

MONTE CARLO CONTROL CARD



Field	Column	Description
1	1 - 8	Not used.
2	9 - 16	SGPDC (program ID)
3	17 - 24	CONTROL (card function)
4	25 - 31	Correction Flags Override $=\Delta$, correct a, e, i, Ω, ω, Mo and Co (if general perturbations) Co is not corrected if no perturbations. Non-blank indicates input correction control as follows: Col. 25 = 1, Correct a 26 = 1, Correct e 27 = 1, Correct i 28 = 1, Correct Ω 29 = 1, Correct ω 30 = 1, Correct Mo 31 = 1, Correct Co A Col $=\Delta$, 0,Do not correct corresponding element.
5	32 - 33	Repeat Counter Override. $=\Delta$, iterate 6 times/element set. = n, iterate n times/element set.

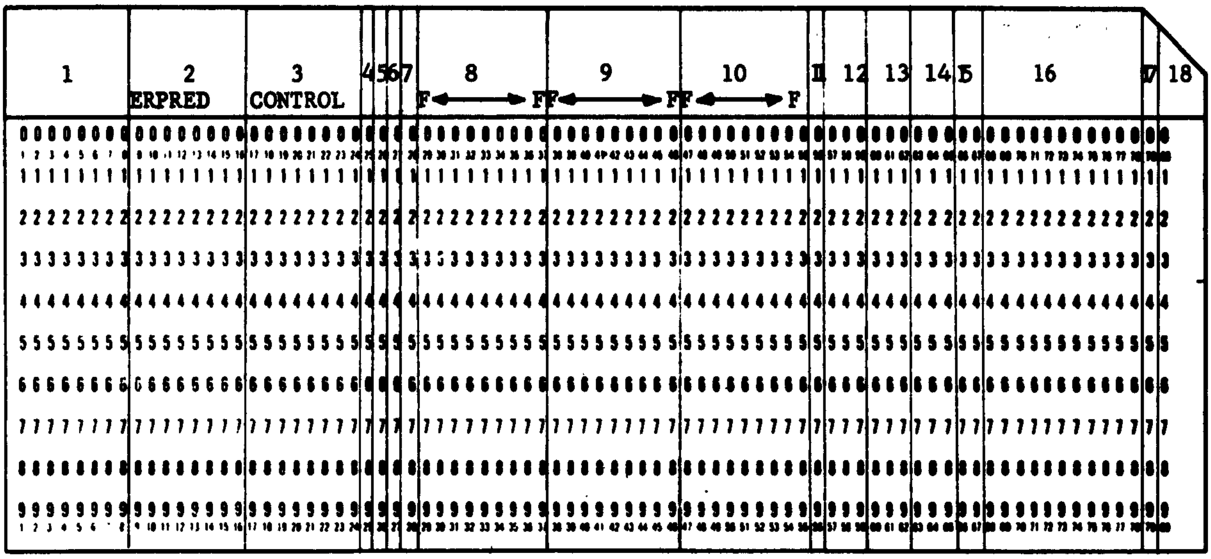
FIGURE 7-13
SGPDC CONTROL CARD

Field	Column	Description
6	34	Perturbations Control = 0 or Δ, no perturbations. = 1, General perturbations
7	35 - 40	Output Options Control. Any column = 0 or Δ indicates print or punch not desired of the corresponding item. Col. 35 = 1, print $\underline{r}$, $\dot{\underline{r}}$ and $\Delta\underline{r}$, $\Delta\dot{\underline{r}}$ Col. 36 = 1, punch $\underline{r}$, $\dot{\underline{r}}$ variant elements Col. 37 = 1, print classical and delta elements Col. 38 = 1, punch classical variant elements Col. 39 = 1, print final RMS values. Print number of residuals accepted and rejected. Col. 40 = 1, print residuals of observations for final variant element set.
8	41 - 43	RMS Multiplier Override. = Δ , RMS multiplier = 1.5 = 0 , no residual rejection. = n , RMS multiplier = n.
9	44 - 48	First Absolute Maximum Override. (absolute maximum for range and angle residuals when the RMS is greater than 50 km.) = Δ , absolute maximum = 1000 km. ≠ Δ , absolute maximum is specified in the field in km.
10	49 - 53	Second Absolute Maximum Override. (absolute maximum for range and angle residuals when the RMS is less than 50 km.) = Δ , second absolute maximum = 75 km. ≠ Δ , second absolute maximum is specified in the field in km.
11	54 - 58	Absolute Maximum for Range Rate, Residuals Override. = Δ , absolute maximum for range rate residuals = .5 km/sec. ≠ Δ , absolute maximum for range rate residuals specified in field in km/sec.
12	59 - 61	Convergence Criterion Override (minimum per cent change in RMS for convergence). = Δ , convergence criterion = 0.05 ≠ Δ , convergence criterion specified in the field.

FIGURE 7-13 cont. SGPDC CONTROL CARD

Field	Column	Description
13	62	Element Errors Statistical Analysis Override: = 0 or Δ, perform the statistical analysis involving the element errors = 1, suppress the computations
14	63-65	S1, the number of standard deviations below the mean at the first cell of the element error frequency distribution. = Δ, S1 set = 3.0 ≠ Δ, S1 specified in the field (floating format).
15	66-68	S2, the number of standard deviations above the mean at the last cell of the element error frequency distribution. = Δ, S2 set = 3.0 ≠ Δ, S2 specified in the field (floating format).
16	69-71	NS, number samples of the element errors used to compute the range of the element error frequency distributions. = Δ, NS set = N (N = total number of delta quantities in the distribution.) ≠ Δ, NS specified in the field (integer format).
17	72-73	I, number of cells in the element error frequency distribution. = Δ, I = $\frac{N+1}{2}$ (integral part) ≠ Δ, I specified in field (integer format)
18	74-80	Not used.

FIGURE 7-13 cont. SGPDC CONTROL CARD



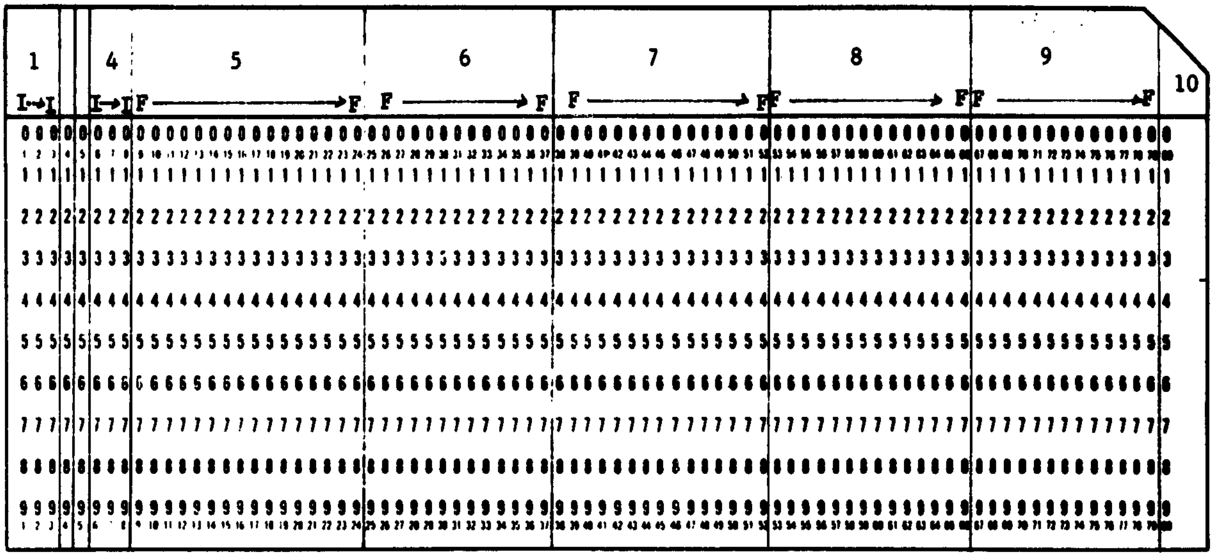
Field	Column	Description
1	1 - 8	Not used.
2	9 - 16	ERPRED (program ID)
3	17 - 24	CONTROL (card function)
4	25	Casetype Option: Δ or 0 - predition points equally spaced in mean anomaly, referenced to the first perigee passage after epoch according to P_i (initial M), ΔP (M interval), P_f (final M) 2 - P_i equal time spaced prediction points in each of four quadrants of mean anomaly referred to the time of the last observation 3 - Prediction points equally spaced in time,referenced to epoch according to P_i (initial time), ΔP (time interval), P_f (final time)
5	26	Coordinate System Option: Δ or 0 - position and velocity errors referred to x, y, z coordinate system ≠ 0 - position and velocity errors referred to $\underline{U}$, $\underline{V}$, $\underline{W}$ coordinate system

FIGURE 7-14

ERPRED CONTROL CARD

Field	Column	Description
6	27	Computation Option: △ or 0 - errors computed using MNREP, with perturbation control as specified on AQTINT control card. 1 - errors computed using MNREP with perturbations 2 - errors computed using MNREP without perturbations 3 - errors computed by multiplying the element errors by partial derivative matrix.
7	28	Output Option: △ or 0 - no optional output ≠ 0 - outputs position and velocity errors in specified coordinate system.
8	29-37	P_i see Field 4 description
9	38-46	ΔP
10	47-55	P_f
11	56	Error Prediction Distribution Override: △ or 0 - distribution computed for position and velocity errors 1 - suppress distribution
12	57-59	S1, multiple of standard deviation, lower limit, for Error Prediction distributions. = △, S1 set = 3.0 ≠ △, S1 specified in field (floating format).
13	60-62	S2, multiple of standard deviation, upper limit, for Error Prediction distributions. = △, S2 set = 3.0 ≠ △, S2 specified in field (floating format)
14	63-65	NS, number of error quantities used to compute the range of distributions for Error Prediction distributions. = △, NS = N (total number of error quantities) ≠ △, NS specified in the field (integer format)
15	66-67	I, number of cells in the distribution for Error Predications distributions = △, $I = \frac{(N+1)}{2}$ (integral part) ≠ △, I specified in the field (integer format)
16	68-78	Not used
17	79	Output option #1 △ or 0 - no optional output ≠ 0 - print error distribution
18	80	Output option #2 △ or 0 - no optional output ≠ 0 - punch error distribution data

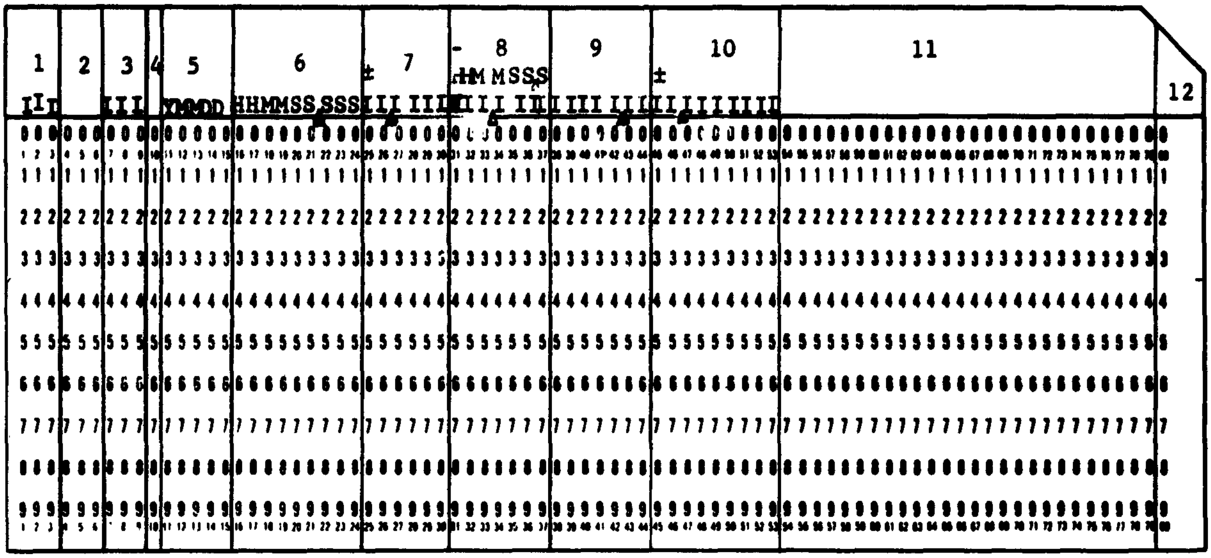
FIGURE 7-14 cont. ERPRED CONTROL CARD



Field	Column	Description
1	1 - 3	Satellite number.
2	4	Not used.
3	5	0 type (generated internally to agree with system code).
4	6 - 8	Sensor number.
5	9 - 24	Time since epoch (minutes).
6	25 - 38	Azimuth (or) right ascension (radians).
7	39 - 52	Elevation (or) declination (radians).
8	53 - 66	Range (Earth radii).
9	67 - 79	Range rate (Earth radii/k_e^{-1}min.)
10	80	0 (card type).

FIGURE 7-15

SYSTEM OBSERVATION CARD



Field	Column	Description
1	1 - 3	Satellite number (for external use only).
2	4 - 6	Not used.
3	7 - 9	Sensor number.
4	10	Not used.
5	11 - 15	YMMDD (Year, Month, Day)
6	16 - 24	HHMMSSSSS (Hour, Minute, Second)
7	25 - 30	Elevation, (or) declination (col 25 can be overpunched for sign) Degrees
8	31 - 37	Azimuth, (or) right ascension Degrees (or) HHMMSSS (Hour, Minute, Second) (Col 31 must be overpunched: + to show azimuth, elevation - to show right ascension, declination)

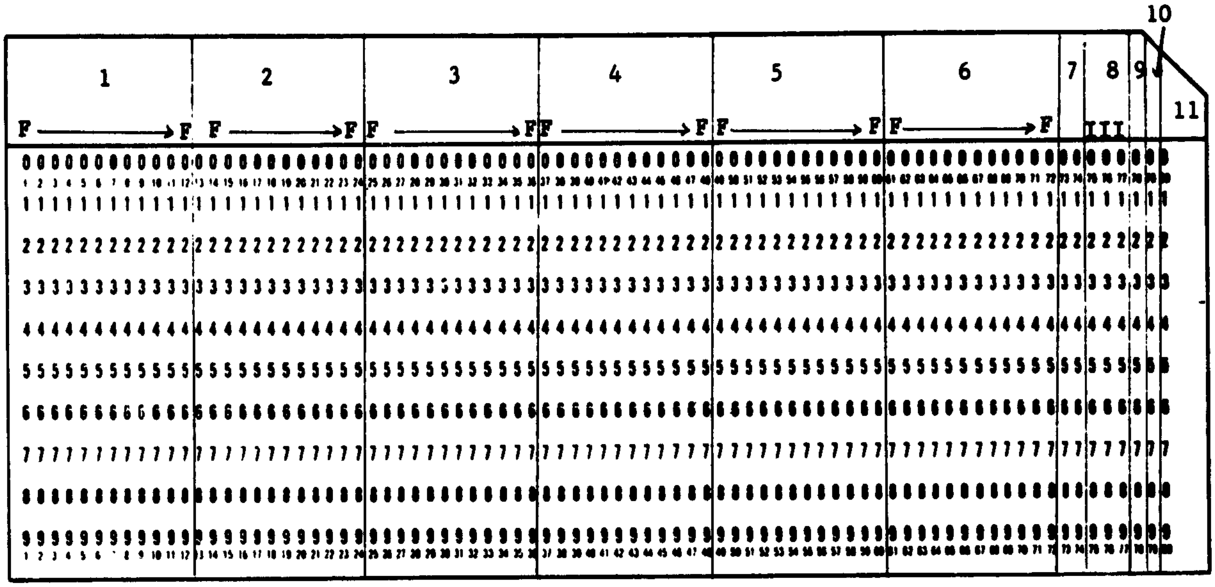
FIGURE 7-16

SPADATS OBSERVATION CARD

Field	Column	Description
9	38-44	Range (kilometers)
10	45-53	Range rate (km/sec)
11	54-79	Not used.
12	80	Card type (= △, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

Note: No decimals on card - they are indicated by carets.

FIGURE 7-16 cont. SPADATS OBSERVATION CARD



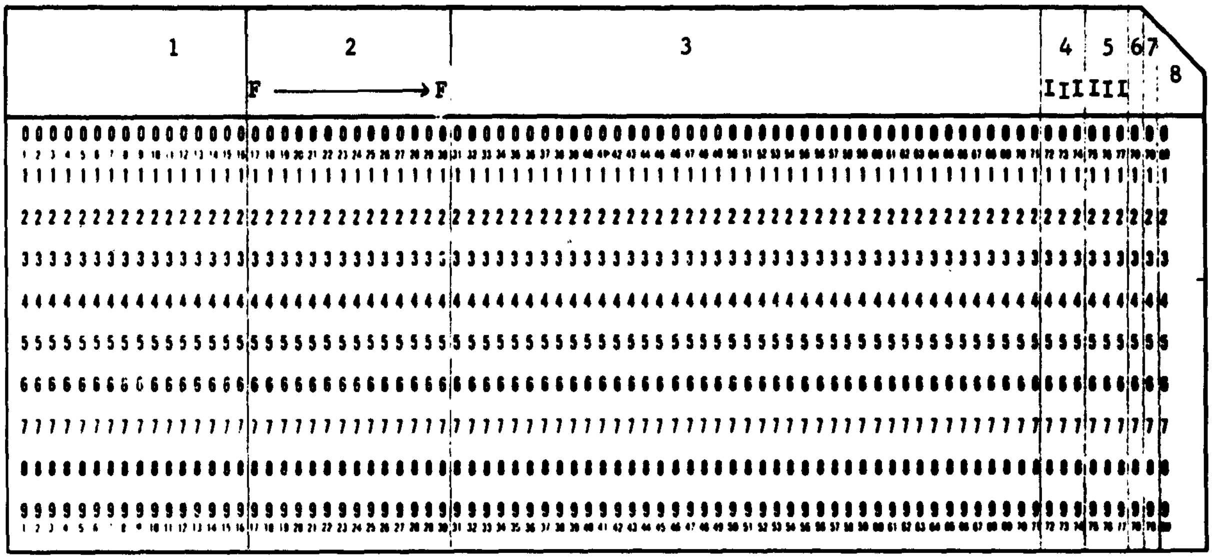
Field	Column	Type A (Classical)	Type B (Position & Velocity)
1	1 - 12	a - Semi-major axis (Earth radii)	x - Equator plane position component along ♈(km)
2	13 - 24	e - Orbital eccentricity	y - Equator plane position component ⊥ to x (km)
3	25 - 36	i - Orbital inclination (deg)	z - Position component orthogonal to x, y (km)
4	37 - 48	Ω - Right ascension of ascending node (deg)	$\dot{x}$ - Equator plane velocity component along ♈(m/sec)
5	49 - 60	ω - Argument of perigee (deg)	$\dot{y}$ - Equator plane to velocity component ⊥ to $\dot{x}$ (m/sec)
6	61 - 72	M_o - Mean anomaly at epoch (deg)	$\dot{z}$ - Velocity component orthogonal to $\dot{x}$, $\dot{y}$ (m/sec)

FIGURE 7-17

VARIANT ELEMENT CARD 1

		Type A	Type B
Field	Column	(Classical)	(Position & Velocity)
7	73-74	Not used.	Not used
8	75-77	Satellite number.	Satellite number.
9	78	Card number = 1	Card number = 1
10	79	Element type = A.	Element type = B.
11	80	Card type = V.	Card type = V.

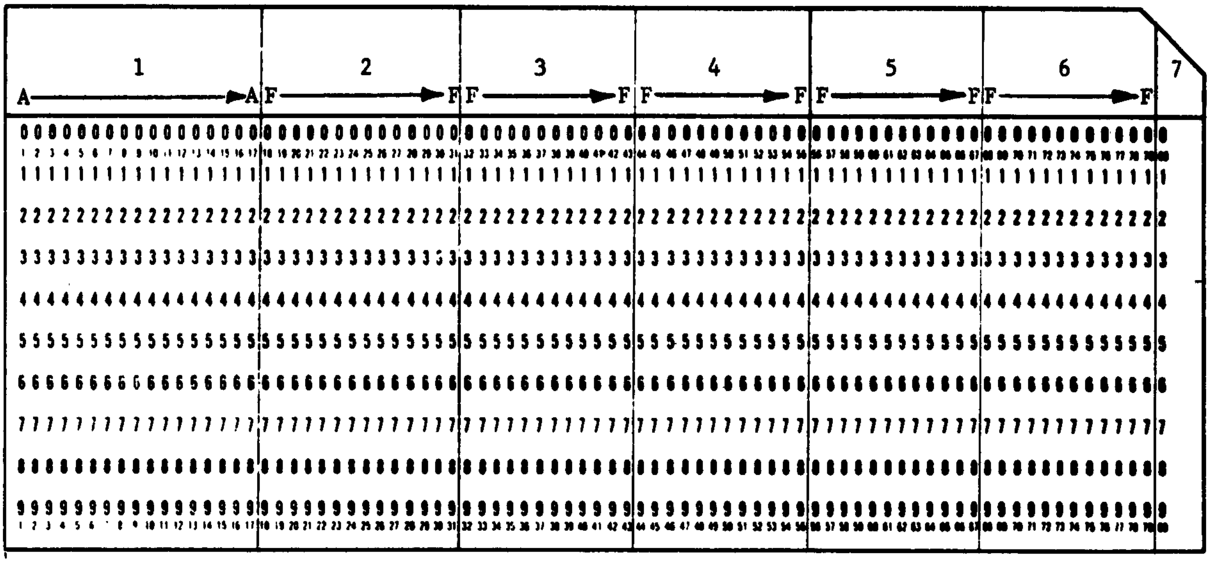
FIGURE 7-17 VARIANT ELEMENT CARD 1



Field	Column	Description
1	1 - 16	Time of epoch (YYMMDDHHMMSS.SSS).
2	17 - 30	$C_{\bullet}$- Rate of change of period (days/rev^2).
3	31 - 71	Not used.
4	72 - 74	Element number.
5	75 - 77	Satellite number.
6	78	Card number - 2.
7	79	Element type = A,B
8	80	Card type = V.

FIGURE 7-18

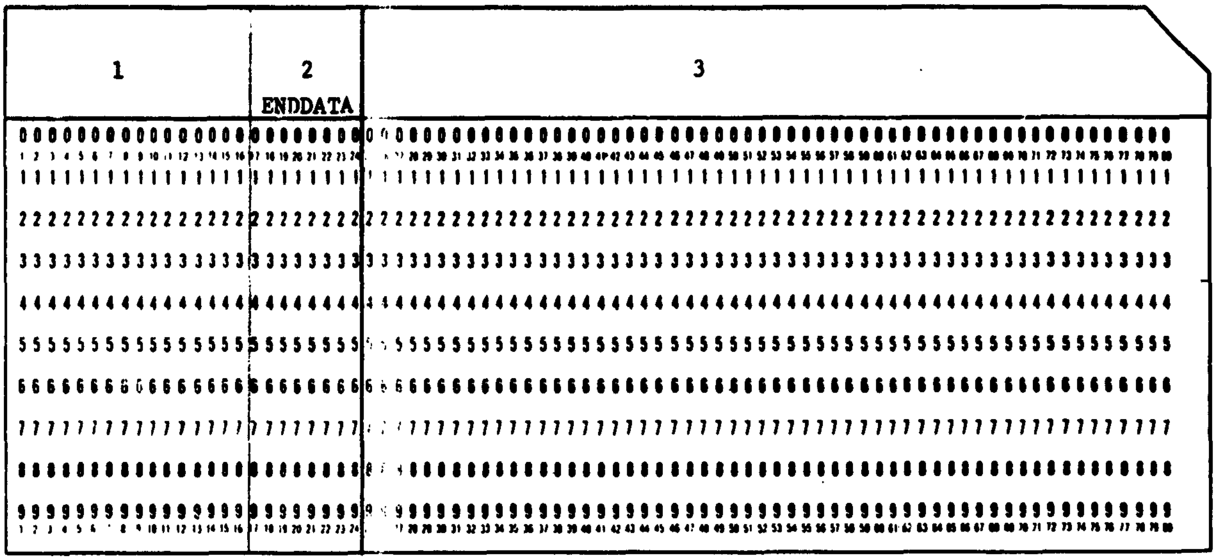
VARIANT ELEMENT CARD 2



Field	Column	Description
1	1 - 17	Name
2	18 - 31	Time since epoch (minutes)
3	32 - 43	Mean
4	44 - 55	Standard deviation
5	56 - 67	Minimum
6	68 - 79	Maximum
7	80	Q (card type)

FIGURE 7-19

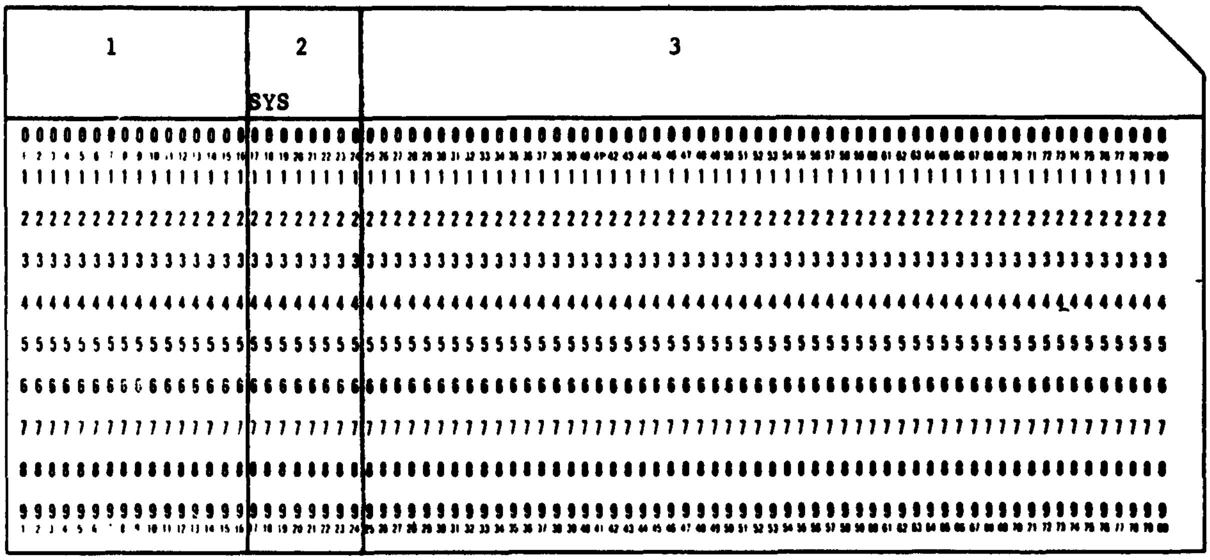
DISTRIBUTION CARD



Field	Column	Description
1	1 - 16	Not used.
2	17 - 24	ENDDATA (card function) defines end of case data.
3	25 - 80	Not used.

FIGURE 7-20

ENDDATA CARD



Field	Column	Description
1	1 - 16	Not used
2	17 - 24	SYS (card function) transfers control to SYS
3	25 - 80	Not used

FIGURE 7-21

SYS CARD

SECTION 8

DICTIONARY

The core location assignments which are common to all of the program units in the Simulation Error Analysis System are defined in the following list. Common block storage assignments are included in the dictionary with their associated lengths and numbers of sets of data specified. The system constants are grouped together, their numerical values being included in the definition.

SIMULATION ERROR ANALYSIS DICTIONARY

A	SEMI-MAJOR AXIS
ADOTU	DOT PRODUCT OF A, D, OR L BAR WITH U BAR
ADOTV	DOT PRODUCT OF A, D, OR L BAR WITH V BAR
ADOTW	DOT PRODUCT OF A, D, OR L BAR WITH W BAR
ALPHA	RT ASCENSION OR AZIMUTH OF OBJ
ALSUN	RIGHT ASCENSION OF THE SUN AT TIME T
AO	A SUB O ,SEMI-MAJOR AXIS AT EPOCH
APMFLAG	DEFINES ERROR PREDICTION COMPUTATIONAL METHOD
AR	A/R ,A=SEMI-MAJOR AXIS ,R=RADIUS VECTOR
ASUBX	X COMP OF A BAR OR A BAR TILDE, UNIT VECTOR PERPENDICULAR TO L BAR
ASUBY	Y COMP OF A BAR OR A BAR TILDE, UNIT VECTOR PERPENDICULAR TO L BAR
ASUBZ	Z COMP OF A BAR OR A BAR TILDE, UNIT VECTOR PERPENDICULAR TO L BAR
ASUXT	A SUB XHO, X COMP WITH RESPECT TO HORIZON SYSTEM OF A BAR TILDE
ASUYT	A SUB YHO, Y COMP WITH RESPECT TO HORIZON SYSTEM OF A BAR TILDE
ASUZT	A SUB ZHO, Z COMP WITH RESPECT TO HORIZON SYSTEM OF A BAR TILDE
AX	X COMPONENT OF VECTOR TO PERIGEE WITH MAG E
AXN	A SUB XN=E*COS OMEGA ,X COMP OF A BAR=E*P BAR IN ORBIT PLANE
AXNO	A SUB XN SUB O=E*COS OMEGA ,X COMP OF A BAR IN EPOCH ORBIT PLANE
AY	Y COMPONENT OF VECTOR TO PERIGEE WITH MAG E
AYN	A SUB YN=E*SIN OMEGA ,Y COMP OF A BAR=E*P BAR IN ORBIT PLANE
AYNO	A SUB YN SUB O=E*SIN OMEGA ,Y COMP OF A BAR IN EPOCH ORBIT PLANE
AZ	Z COMPONENT OF VECTOR TO PERIGEE WITH MAG E
BETA	ANGLE BETWEEN ORBITAL PLANE AND PLANE OF THE OBJ AS OBSERVED
BIASALP	BIAS ALPHA IN RAD. (EITHER RIGHT ASCENSION OR AZIMUTH)
BIASDLT	BIAS DELTA IN RAD. (EITHER DECLINATION OR ELEVATION)
BIASRHO	BIAS RANGE IN ER
BIASRRI	BIAS RANGE RATE IN ER/KEMIN
C	C DOUBLE PRIME=1/2*DN/DT ,A DRAG COEFF
CAPPF	DEFINES ERROR PREDICTION FINAL PREDICTION POINT
CAPPI	DEFINES ERROR PREDICTION INITIAL PREDICTION POINT
CAPR	MAGNITUDE OF INERTIAL VECTOR FROM STATION
CAPX	X COMP OF CAPITAL R BAR =STATION VECTOR
CAPY	Y COMP OF CAPITAL R BAR =STATION VECTOR

```
CAPZ       Z COMP OF CAPITAL R BAR =STATION VECTOR
CARDS      INDICATOR OF DATA CARDS FOUND
CCARD      INDICATOR OF CONTROL CARDS FOUND
CASETYP    DEFINES ERROR PREDICTION POINTS
C0         C SUB 0=1/2*DP/DN ,A DRAG COEFF
COBLOC     CONTAMINATED OBSERVATION BUFFER, 3001 CELLS (500*6 WORD ENTRIES+1
           WORD)
ICHARCT    NO. OF CHAR. IN LAST VARIABLE FIELD PARAMETER, FIX INTEGER T15.
COBSTB     RIGHT ADDRESS = CURRENT LOCATION BEING USED IN COBLOC
CONBITS    SYS MASTER CONTROL SWITCH
CONLINE    10 WORD INPUT CARD IMAGE BUFFER IN SYS
COCOUNT    RIGHT ADDRESS = NUMBER OF 6 WORD ENTRIES IN COBLOC
COSEO      COS(E+OMEGA) ,E=ECCENTRIC ANOM ,OMEGA=ARG OF PERIFOCUS
COSI       COS I ,I=INCLINATION
COSO       COS NODE =N SUB X
COSOM      COS OMEGA ,COSINE OF ARG OF PERIFOCUS
COSPH      COS LATITUDE
COSPHIS    COSINE OF PHI SUB S, GEOCENTRIC LATITUDE OF SUB-SATELLITE POINT
COSTH      COS THETA ,THE SIDEREAL TIME
COSU       COS ARG OF LATITUDE
COSVO      COS TRUE ANOMALY AT EPOCH
CSALS      COSINE OF RT. ASCEN. OF SUN AT TIME T
CSDLS      COSINE OF DECLINATION OF THE SUN AT TIME T
CXDOT      X COMP OF CAP R BAR DOT (STATION VELOCITY VECTOR)(RAD/KEMIN)
CYDOT      Y COMP OF CAP R BAR DOT (STATION VELOCITY VECTOR)(RAD/KEMIN)
CZDOT      Z COMP OF CAP R BAR DOT (STATION VELOCITY VECTOR)(RAD/KEMIN)
C1T        C1*T
C3         CORRECTIVE QUANTITY, USED IN COMPUTING TRUE LONGITUDE OF SUN
DCCCP1     DC CONTROL FOR ELEMENT CORRECTION SPECIFICATION
DCCCP10    DC CONTROL FOR DISTRIBUTION SUPPRESSION
DCCCP2     DC CONTROL FOR NUMBER OF ITERATIONS
DCCCP3     DC PERTURBATION CONTROL
DCCCP4     DC OUTPUT OPTION CONTROL
DCCCP5     DC RMS FACTOR CONTROL
DCCCP6     DC CONTROL FOR FIRST ABSOLUTE MAX RANGE AND ANGLE (KM)
DCCCP7     DC CONTROL FOR SECOND ABSOLUTE MAX RANGE AND ANGLE (KM)
```

DCCCP8	DC CONTROL FOR ABSOLUTE MAX RANGE RATE (KM/SEC)
DCCCP9	DC CONTROL FOR CONVERGENCE CRITERION
DCFIRST	INITIALIZATION SWITCH FOR SGPDC
DCI	I FOR DC DISTRIBUTION (NO. OF CELLS IN DISTRIBUTION)
DCNS	NS FOR DC DISTRIBUTION (NO. OF QUANTITIES TO DEFINE DISTRIBUTION)
DCS1	S1 FOR DC DISTRIBUTION (NO. OF ST. DEV. BELOW THE MEAN)
DCS2	S2 FOR DC DISTRIBUTION (NO. OF ST. DEV. ABOVE THE MEAN)
DEBLOC	DELTA ELEMENT BUFFER, 701 CELLS (100*7 WORD ENTRIES + 1 WORD)
DECOUNT	RIGHT ADDRESS = NUMBER OF 7 WORD ENTRIES IN DEBLOC
DELA	SAME AS DELAO
DELAO	TOTAL CORRECTION IN SEMI-MAJOR AXIS AT EPOCH
DELCAPP	DEFINES INTERVAL BETWEEN ERROR PREDICTIONS
DELCO	TOTAL CORRECTION IN DRAG PARAMETER AT EPOCH
DELE	SAME AS DELEO
DELEO	TOTAL CORRECTION IN ECCENTRICITY AT EPOCH
DELI	SAME AS DELIO
DELIO	TOTAL CORRECTION IN INCLINATION AT EPOCH
DELMO	TOTAL CORRECTION IN MEAN ANOMALY AT EPOCH
DELNOD	SAME AS DELNODO
DELNODO	TOTAL CORRECTION IN LONGITUDE OF NODE AT EPOCH
DELOM	SAME AS DELOMO
DELOMO	TOTAL CORRECTION IN ARG OF PERIGEE AT EPOCH
DELSTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN DEBLOC
DELTA	DECLINATION OR ELEVATION OF OBJ
DELTX	X COMP OF DELTA L BAR = L BAR OBS - L BAR COMPUTED
DELTY	Y COMP OF DELTA L BAR = L BAR OBS - L BAR COMPUTED
DELTZ	Z COMP OF DELTA L BAR = L BAR OBS - L BAR COMPUTED
DENM	(1+(1-E**2)**1/2)**2 * (1-E**2)**1/2 ,E =ECCENTRICITY
DIRF	DEFINES ANTI OR POST MERIDIAN POSITION OF HF
DIRI	DEFINES ANTI OR POST MERIDIAN POSITION OF HI
DLSUN	DECLINATION OF SUN AT TIME T
DPCOUNT	RIGHT ADDRESS = NUMBER OF 8 WORD ENTRIES IN DPVBLOC
DOLS	BCD DOLLAR SIGN T47, LEADING ZEROS.
DPVBLOC	DELTA POS-VEL BUFFER, 801 CELLS (100*8 WORD ENTRIES + 1 WORD)
DPVFLAG	DEFINES ERROR PREDICTION OUTPUT
DPVSTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN CPVBLOC

DSUBX	X CØMP ØF D BAR ØR D BAR TILDE, UNIT VECTØR PERPENDICULAR TØ L BAR
DSUBY	Y CØMP ØF D BAR ØR D BAR TILDE, UNIT VECTØR PERPENDICULAR TØ L BAR
DSUBZ	Z CØMP ØF D BAR ØR D BAR TILDE, UNIT VECTØR PERPENDICULAR TØ L BAR
DSUXT	D SUB XHØ, X CØMP WITH RESPECT TØ HØRIZØN SYSTEM ØF D BAR TILDE
DSUYT	D SUB YHØ, Y CØMP WITH RESPECT TØ HØRIZØN SYSTEM ØF D BAR TILDE
DSUZT	D SUB ZHØ, Z CØMP WITH RESPECT TØ HØRIZØN SYSTEM ØF D BAR TILDE
E	ECCENTRICITY
EBLØC	NØMINAL ELEMENT BUFFER, 13 CELLS (1*12 WØRD ENTRY + 1 WØRD)
ECØSE	E*CØS E =ECCENTRICITY * CØS ECCENTRIC AÑØM
ECØSV	E*CØS V, ECCENTRICITY*CØS TRUE ANØMALY
ECØUNT	RIGHT ADDRESS = NUMBER ØF 12 WØRD ENTRIES IN EBLØC
ELMSTB	RIGHT ADDRESS = CURRENT LØCATIØN BEING USED IN VEBLØC
ELMTYP	BCD ELEMENT TYPE FRØM INPUT CARD (0000000X' X=A,B,C ØR D)
ELNØ	ELEMENT NØ, IDENTIFIES VARIANT ELEMENT SETS
ENTRYJA	LEFT ADDRESS CØNTAINS JA AT ENTRY TØ SYS
EØ	E SUB O ,ECCENTRICITY AT EPØCH
EPØCH	TIME ØF EPØCH
ERPDØPT	ERPRED DISTRIBUTIØN ØPTIØN
ERPI	I FØR ERRØR PRED. DIST. (NØ. ØF CELLS IN DISTRIBUTIØN)
ERPNS	NS FØR ERRØR PRED. DIST. (NØ. ØF QUANTITIES TØ DEFINE DISTRIBUTIØN)
ERPS1	S1 FØR ERRØR PRED. DIST. (NØ. ØF ST. DEV. BELØW THE MEAN)
ERPS2	S2 FØR ERRØR PRED. DIST. (NØ. ØF ST. DEV. ABØVE THE MEAN)
ESINE	E*SIN E =ECCENTRICITY * SIN ECCENTRIC ANØM
ESINEV	E*SIN V, ECCENTRICITY*SIN TRUE ANØMALY
ESQ	E**2 = ECCENTRICITY SQUARED
ESUBX	X CØMP ØF E BAR = UNIT VECTØR FRØM ØBS TØ EAST
ESUBY	Y CØMP ØF E BAR = UNIT VECTØR FRØM ØBS TØ EAST
ESUBZ	Z CØMP ØF E BAR = UNIT VECTØR FRØM ØBS TØ EAST
EXSWTCH	EXECUTIVE SWITCH' BIT 0=NØ-RUN BIT, BIT 1=CASE CARD BIT
HF	FINAL ELEVATIØN ANGLE FØR SENSØR
HI	BEGINNING ELEVATIØN ANGLE FØR SENSØR
HXØ	H SUB X SUB O,X CØMP ØF H BAR (ANGULAR MØMENTUM VECT) AT EPØCH
HYØ	H SUB Y SUB O,Y CØMP ØF H BAR (ANGULAR MØMENTUM VECT) AT EPØCH
HZØ	H SUB Z SUB O,Z CØMP ØF H BAR (ANGULAR MØMENTUM VECT) AT EPØCH
IBLØC	INITIALIZATIØN PARAMETERS BUFFER, 701 CELLS (100*7 WØRD ENTRIES+1 WØRD)

ICOUNT	RIGHT ADDRESS = NUMBER OF 7 WORD ENTRIES IN IBLOC
IDELTAT	OBSERVATION TIME INTERVAL
ISTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN IBLOC
K	NUMBER OF MOCARO-DC LOOPS REQUESTED ON MOCARO CONTROL CARD-(NO OF RUNS)
KCOUNT	COUNTER INDICATING NO. OF SUCCESSFUL MOCARO-DC LOOPS COMPLETED
KEJGRAV	((P SUB 0)**(7/2))/((K SUB E)*(J)*((A SUB E)**2))=PZERO**2*RTP/XJGRCF
LCOBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN COBLOC
LDEBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN DEBLOC
LDPBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN DPVBLOC
LEBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN EBLOC
LIBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN IBLOC
LLO	TRUE LONGITUDE AT EPOCH (RADIANS)
LOBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN OBLOC
LOCCOB	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN COBLOC
LOCDEL	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN DEBLOC
LOCDPV	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED ~~IN DPVBLOC~~
LOCELM	R ADR = BEG LOC' L ADR = FIRST LOC ~~NOT~~ USED IN VEBLOC
LOCI	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN IBLOC
LOCOBS	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN OBLOC
LOCPOB	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN POBLOC
LOCSAT	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN EBLOC
LOCSEN	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN SBLOC
LOCSIG	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN SGBLOC
LOCSTP	R ADR = BEG LOC' L ADR = FIRST LOC NOT USED IN STPBLOC
LPOBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN POBLOC
LSBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN SBLOC
LSGBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN SGBLOC
LSTBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN STPBLOC
LTBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN TBLOC
LVEBLOC	R ADR = BEG LOC' L ADR = MAX LOC ALLOWABLE IN VEBLOC
MCFIRST	INITIALIZATION SWITCH FOR MOCARO
MM	BCD MINUTES AND SECONDS OF EPOCH (MMSS.SSS)
NUSEG	NUMBER OF SEGMENTS IN SATELLITE PASS
OALT	STATION ALTITUDE
OBCOUNT	RIGHT ADDRESS = NUMBER OF 6 WORD ENTRIES IN OBLOC

OBLOC	SIMULATED OBSERVATION BUFFER, 3001 CELLS (500*6 WORD ENTRIES + 1 WORD)
OBSDUR	OBSERVATION DURATION
OBSOPT	OBSERVATION SIMULATION OUTPUT CONTROL
OBSR	MAGNITUDE OF R BAR, VECTER DIRECTED TO OBJECT
OBSSTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN OBLOC
OMEGA	ARGUMENT OF PERIFOCUS
OMGAS	OMEGA SUB S, CHANGE IN ARGUMENT OF PERIFOCUS SINCE EPOCH
OMGDT	D(OMEGA)/DT ,RATE OF CHANGE OF ARG OF PERIFOCUS
OPOUT1	OUTPUT OPTION FOR ERPRED DISTRIBUTION - HISTOGRAMS
OPOUT2	OUTPUT OPTION FOR ERPRED DISTRIBUTION - PUNCHED CARDS
OTPFLAG	DEFINES ERROR PREDICTION REF COORDINATE SYSTEM
OTYPE	DEFINES OBSERVATION TYPE FROM A STATION
P	SEMI-LATUS RECTUM = PARAMETER
PERT	PERTURBATION CONTROL EXCLUDING DC
PHIRD	STATION LATITUDE =PHI (USUALLY IN RADIANS)
PN	NODAL PERIOD
PNCHOBS	OUTPUT OPTION FOR OBSERVATION SIMULATION - PUNCHED CARDS
POBLOC	PROCESSED OBSERVATION BUFFER, 9500 CELLS (500*19 WORD ENTRIES-VARIABLE)
POBSTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN POBLOC
POCOUNT	RIGHT ADDRESS = NUMBER OF 19 WORD ENTRIES IN POBLOC
PROGNO	PROGRAM NUMBER CURRENTLY RUNNING, FIXED INTEGER T15.
PSI	ANGLE PSI
PX	X COMPONENT OF UNIT VECTOR TO PERIGEE
PY	Y COMPONENT OF UNIT VECTOR TO PERIGEE
PZ	Z COMPONENT OF UNIT VECTOR TO PERIGEE
Q	PERIFOCAL DISTANCE AT TIME T
QO	Q SUB O ,PERIFOCAL DISTANCE AT EPOCH
QX	X COMPONENT OF UNIT VECTOR NORMAL TO P BAR
QY	Y COMPONENT OF UNIT VECTOR NORMAL TO P BAR
QZ	Z COMPONENT OF UNIT VECTOR NORMAL TO P BAR
R	RADIUS VECTOR
RANGE	OBS RANGE
RDOT	RATE OF CHANGE OF RADIUS VECTOR
REFYEAR	YEAR OF INITIAL ELEMENT EPOCH
RHOC	RHO SUB C, COMPUTED RANGE

RHODT	RHO DOT, COMPUTED RANGE RATE
RHOX	X COMP OF RANGE VECT (RHO BAR) =XI
RHOY	Y COMP OF RANGE VECT (RHO BAR) =ETA
RHOZ	Z COMP OF RANGE VECT (RHO BAR) =ZETA
RHOO	INITIAL RANGE FOR ELEMENT TYPE D
RODOT	RANGE RATE OF OBJ (OBSERVED)
RODTX	X COMP OF RANGE RATE VECT (RHO BAR DOT) = XI DOT
RODTY	Y COMP OF RANGE RATE VECT (RHO BAR DOT) = ETA DOT
RODTZ	Z COMP OF RANGE RATE VECT (RHO BAR DOT) = ZETA DOT
ROMAX	MAXIMUM RANGE ALLOWABLE FOR SENSOR
RSQ	R**2 = RADIUS VECTOR SQUARED
RTA	A**1/2 ,SQUARE ROOT OF SEMI-MAJOR AXIS
RTESQ	(1-E**2)**1/2 WHERE E IS THE ECCENTRICITY
RTP	P**1/2 =SQ ROOT OF THE PARAMETER
RTPRTMU	SQUARE ROOT OF MU*SEMI-LATUS RECTUM
RVDOT	R*V DOT =RADIUS VECTOR * RATE OF CHANGE OF TRUE ANOMALY
SATN	SATELLITE NO, IDENTIFIES THE INITIAL ELEMENT SET
SBLOC	SENSOR BUFFER, 601 CELLS (100*6 WORD ENTRIES + 1 WORD)
SCOUNT	RIGHT ADDRESS = NUMBER OF 6 WORD ENTRIES IN SBLOC
SEGNOS	SEGMENT OF SATELLITE PASS CHOSEN FOR OBSERVATION
SENSTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN SBLOC
SGBLOC	ST DEV AND BIAS BUFFER, 901 CELLS (100*9 WORD ENTRIES + 1 WORD)
SGCOUNT	RIGHT ADDRESS = NUMBER OF 9 WORD ENTRIES IN SGBLOC
SIGALP	SIGMA ALPHA IN RAD. (EITHER RIGHT ASCENSION OR AZIMUTH)
SIGDLT	SIGMA DELTA IN RAD. (EITHER DECLINATION OR ELEVATION)
SIGRHO	SIGMA RANGE IN ER
SIGRRT	SIGMA RANGE RATE IN ER/KEMIN
SIGSTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN SGBLOC
SINEO	SIN(E+OMEGA) ,E=ECCENTRIC ANOM ,OMEGA=ARG OF PERIFOCUS
SINI	SIN I ,I=INCLINATION
SINO	SIN NODE =N SUB Y
SINOM	SINE OF OMEGA, ARGUMENT OF PERIFOCUS
SINPH	SIN LATITUDE
SINPHIS	SINE OF PHI SUB S, GEOCENTRIC LATITUDE OF THE SUB-SATELLITE POINT
SINTH	SIN THETA ,THE SIDEREAL TIME
SINU	SIN ARG OF LATITUDE

SINV0	SIN TRUE ANOMALY AT EPOCH
SNALS	SINE OF RT ASCEN OF SUN AT TIME T
SNHSN	SINE OF ELEVATION OF SUN AT TIME T
SSUBX	X COMP OF S BAR, UNIT VECTOR FROM OBS TO SOUTH
SSUBY	Y COMP OF S BAR, UNIT VECTOR FROM OBS TO SOUTH
SSUBZ	Z COMP OF S BAR, UNIT VECTOR FROM OBS TO SOUTH
STAID	STATION IDENTIFICATION
SATSTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN EBLOC
STCOUNT	RIGHT ADDRESS = NUMBER OF 24 WORD ENTRIES IN STPBLOC
STPBLOC	ST PROC DELTA POS-VEL BUFFER, 961 CELLS (40*24 WORD ENTRIES + 1 WORD)
STPSTB	RIGHT ADDRESS = CURRENT LOCATION BEING USED IN STPBLOC
SUNLX	X COMP OF SUN L BAR =OBS UNIT VECTOR OBS TO SUN
SUNLY	Y COMP OF SUN L BAR =OBS UNIT VECTOR OBS TO SUN
SUNLZ	Z COMP OF SUN L BAR =OBS UNIT VECTOR OBS TO SUN
SXJGRCF	DC STORAGE FOR XJGRCF
T	TIME SINCE EPOCH (MIN)
TBLOC	TIME BUFFER, 1201 CELLS (VARIABLE ENTRY LENGTH)
THGR0	GREENWICH SIDEREAL TIME AT BEGINNING OF EPOCH YEAR
THGRT0	GREENWICH SID TIME AT EPOCH (RADIANS)
THTA	THETA =SIDEREAL TIME
T0	SAME AS EPOCH
TOBF	FINAL ERROR PREDICTION TIME
TOPT	ACQUISTION TIME OUTPUT CONTROL
U	MEAN ARG OF LATITUDE
UMU0	U-U AT EPOCH, U= MEAN ARGUMENT OF LATITUDE
U0	U SUB 0 ,MEAN ARG OF LAT AT EPOCH
UX	U SUB X ,X COMP OF U BAR=UNIT VECTOR DIRECTED TO OBJ
UY	U SUB Y ,Y COMP OF U BAR=UNIT VECTOR DIRECTED TO OBJ
UZ	U SUB Z ,Z COMP OF U BAR=UNIT VECTOR DIRECTED TO OBJ
VDOT	RATE OF CHANGE OF TRUE ANOMOLY
VEBLOC	VARIANT ELEMENT BUFFER, 3001 CELLS (500*6 WORD ENTRIES + 1 WORD)
VECOUNT	RIGHT ADDRESS = NUMBER OF 7 WORD ENTRIES IN VEBLOC
VELNO	VARIANT ELEMENT SET NO.
VMAG	VECTOR MAGNITUDE OF RESIDUALS OF OBSERVATION
VTYPE	DEFINES STATION AS VISUAL OR NON-VISUAL TYPE (0000000X' X=V OR =V)
VX	V SUB X ,X COMP OF V BAR=UNIT VECTOR IN ORB PLANE PERPEN TO U BAR

```
VY        V SUB Y ,Y COMP OF V BAR=UNIT VECTOR IN ORB PLANE PERPEN TO U BAR
VZ        V SUB Z ,Z COMP OF V BAR=UNIT VECTOR IN ORB PLANE PERPEN TO U BAR
WX        W SUB X ,X COMP OF W BAR=UNIT VECTOR PERPENDICULAR TO ORBIT PLANE
WY        W SUB Y ,Y COMP OF W BAR=UNIT VECTOR PERPENDICULAR TO ORBIT PLANE
WZ        W SUB Z=COS I, Z COMP OF W BAR=UNIT VECTOR PERPENDIC TO ORB PLANE
W1        FIRST WORD OF TWO WORD BUFFER CONTAINING IMAGE OF VARIABLE FIELD PARM
W2        SECOND WORD OF TWO WORD BUFFER CONTAINING IMAGE OF VARIABLE FIELD PARM
X         X COMP OF R BAR=VECTOR DIRECTED TO OBJ
XDOT      X COMP OF R BAR DOT
XKELSO    K SUB E * L GRAVE SUB SO ,USED IN COMPUTING L
XL        MEAN LONGITUDE
XLAMBA    STATION LONGITUDE
XLO       L SUB O ,MEAN LONG AT EPOCH
XLSUBX    L SUB XO, X COMP OF L BAR SUB O, OBS UNIT VECTOR FROM OBS.TO OBJ
XLSUBY    L SUB YO, Y COMP OF L BAR SUB O, OBS UNIT VECTOR FROM OBS.TO OBJ
XLSUBZ    L SUB ZO, Z COMP OF L BAR SUB O, OBS UNIT VECTOR FROM OBS.TO OBJ
XLSUNO    MEAN LONGITUDE OF SUN AT START OF EPOCH YEAR
XLSUNT    MEAN LONGITUDE OF THE SUN AT TIME T
XLSUXH    L SUB XHO ,L BAR SUB HO=OBS UNIT VECT FROM OBS TO OBJ-WRT HORIZON
XLSUYH    L SUB YHO ,L BAR SUB HO=OBS UNIT VECT FROM OBS TO OBJ-WRT HORIZON
XLSUZH    L SUB ZHO ,L BAR SUB HO=OBS UNIT VECT FROM OBS TO OBJ-WRT HORIZON
XLX       X COMP OF COMP L BAR ,UNIT VECT FROM OBS TO OBJ
XLY       Y COMP OF COMP L BAR ,UNIT VECT FROM OBS TO OBJ
XLZ       Z COMP OF COMP L BAR ,UNIT VECT FROM OBS TO OBJ
XMO       MEAN ANOMOLY AT EPOCH
XMX       M SUB X ,X COMP OF M BAR=UNIT VECT IN ORB PLANE PERPENDICULAR TO N BAR
XMY       M SUB Y ,Y COMP OF M BAR=UNIT VECT IN ORB PLANE PERPENDICULAR TO N BAR
XMZ       M SUB Z=SIN I= Z COMP OF M BAR=UNIT VECT IN ORB PLANE PERP TO N BAR
XN        MEAN ANGULAR MOTION
XNO       N SUB O ,MEAN ANGULAR MOTION AT EPOCH
XNODE     LONG OF THE ASCENDING NODE
XNODEO    NODE SUB O ,LONG OF ASCENDING NODE AT EPOCH
XNODOT    D(NODE)/DT ,RATE OF CHANGE OF LONG OF THE ASCENDING NODE
XNX       X COMP OF N BAR (UNIT VECT DIRECTED TOWARD ASCEN NODE) =SIN NODE
XNY       Y COMP OF N BAR (UNIT VECT DIRECTED TOWARD ASCEN NODE) =COS NODE
XNZ       Z COMP OF N BAR (UNIT VECT DIRECTED TOWARD ASCEN NODE) =O
```

```
XOBS      X COMP OF OBS R BAR, VECTOR DIRECTED TO OBJ
XOVCT     CAP X/COS THETA =CAP Y/SIN THETA ,USED IN COMPUTING STATION VECT
Y         Y COMP OF R BAR=VECTOR DIRECTED TO OBJ
YDOT      Y COMP OF R BAR DOT
YOBS      Y COMP OF OBS R BAR, VECTOR DIRECTED TO OBJ
YY        BCD YEAR, MONTH, DAY, HOUR OF EPOCH (YYMMDDHH)
Z         Z COMP OF R BAR=VECTOR DIRECTED TO OBJ
ZDOT      Z COMP OF R BAR DOT
ZOBS      Z COMP OF OBS R BAR, VECTOR DIRECTED TO OBJ
ZSUBX     X COMP OF Z BAR, UNIT VECTOR FROM OBS TO ZENITH
ZSUBY     Y COMP OF Z BAR, UNIT VECTOR FROM OBS TO ZENITH
ZSUBZ     Z COMP OF Z BAR, UNIT VECTOR FROM OBS TO ZENITH
                              CONSTANTS
C1        .01720279          = ASTRO. CONSTANT USED IN COMPUTING SUN'S POS. RAD/
                               DAY
C2        .033358            = ASTRO. CONSTANT USED IN COMPUTING SUN'S POS. RAD
C4        .043109            = ASTRO. CONSTANT USED IN COMPUTING SUN'S POS. RAD
C5        .4336635           = ASTRO. CONSTANT USED IN COMPUTING SUN'S POS. RAD
DAPKE     .009338231         = DAYS/KEMIN
DDPRM     82505.9225         = (DEG/DAY)/(RAD/MIN)
DE2RA     .0174532925        = RADIANS/DEGREE
EPSQD     .66934216E-3       = 2*F-F**2, F=FLATTENING OF EARTH
ERK2KMS   7.905              = (KILOMETERS/SECOND)/(EARTH RADII/KEMIN)
ERPKM     .15678527E-3       = EARTH RADII/KILOMETER
ERPMI     .156785275E-6      = EARTH RADII/METER
F         .33523299E-2       = F, FLATTENING OF THE EARTH F=1/298.3
FL00      0                  = 0
FL01      1                  = 1
FL010     10                 = 10
FL0100    100                = 100
FL02      2                  = 2
FL03      3                  = 3
FL04      4                  = 4
FL05      5                  = 5
FL06      6                  = 6
FL07      7                  = 7
```

```
FLØ8       8                  = 8
FLØ9       9                  = 9
FLØ90      90                 = 90
FLP25      .25068448          = RØTATIØN RATE ØF THE EARTH IN DEG/SØLAR MINUTE
FL100      100                = (SAME AS FLØ100)
FL180      180                = 180
FL300      300                = 300
FL360      360                = 360
FL600      600                = 600
F1000      1000               = 1000
HALF       .5                 = 1/2
KMS2ERK    .1265022138        = (EARTH RADII/KEMIN)/(KILØMETERS/SECØND)
MSPRK      7905.              = (METERS/SECØND)/(EARTH RADII/KEMIN)
MS2ERK     .1265022138E-3     = (EARTH RADII/KEMIN)/(METERS/SECØND)
MTPER      6378150            = METERS/EARTH RADII
MTS2ERK    .1265022138E-3     = (SAME AS MS2ERK)
PI         3.1415926535       = PI
PIØV2      1.5707963267       = (SAME AS PIØ2)
PIØV4      .785398162         = PI/4
PIØ2       1.5707963267       = PI/2
PIØ36      -.02741556771      = -(PI**2)/360
PISQ       9.8696044          = PI**2
PØ         84.490322          = PERIØD ØF A SURFACE SATELLITE (A=1) IN MINUTES
P1HL       PLUS ØNE HALF (F BIT) IN THE LEFT ADDRESS
P1LADR     PLUS 1 IN THE LEFT ADDRESS
P2LADR     PLUS 2 IN THE LEFT ADDRESS
P3JAØ2     1.62339E-3         = 3*(J SUB 2)*(A SUB E)**2/2  =J'
P3LADR     PLUS 3 IN THE LEFT ADDRESS
P4LADR     PLUS 4 IN THE LEFT ADDRESS
P5LADR     PLUS 5 IN THE LEFT ADDRESS
P6LADR     PLUS 6 IN THE LEFT ADDRESS
P7LADR     PLUS 7 IN THE LEFT ADDRESS
P7RADR     PLUS 7 IN THE RIGHT ADDRESS
P8LADR     PLUS 8 IN THE LEFT ADDRESS
P9LADR     PLUS 9 IN THE LEFT ADDRESS
QTR        .25                = 1/4
```

RADYN	57.29577951	= DEGREES/RADIAN
RPTIM	.43752691E-2	= RØTATIØN RATE ØF EARTH IN RADIANS/SØLAR MINUTE
SIDRT	.9856472	= RØTATIØN RATE ØF EARTH IN DEG/SØLAR DAY - 360 DEG.
SIDRT+1	360.9856472	= RØTATIØN RATE ØF EARTH IN DEG/SØLAR DAY - 360 DEG.
SQRMU	1	= (SAME AS SQRTMU)
SQRTMU	1	= MU**(1/2)' MU=SUM ØF EARTH + SAT MASSES/EARTH'S MASS
STPER	3963.199	= STATUTE MILES/EARTH RADII
TENM6	.000001	= 1*(10**-6)
THDØT	.05883447	= RØTATIØN RATE ØF THE EARTH IN RADIANS/KEMIN
TWØPI	6.2831853	= 2*PI
XJGRCF	.120717162E-3	= (K SUB E)*J*(A SUB E)**2
XKE	.07436574	= K SUB E
XKERTM	.07436574	= (K SUB E)*MU**(1/2)
XKMPER	6378.150	= KILØMETERS/EARTH RADII
XKNØTS	15366.602	= (NMI/HR)/(EARTH RADII/KEMIN)
XMFLØ1	-1	= -1
XMNPDA	1440	= MINUTES/DAY
XM2Ø3	-.6666666667	= -2/3
XM27Ø4	-6.75	= -27/4
XM3Ø2	-1.5	= -3/2
XM4Ø3	-1.333333333	= -4/3
XNMPER	3445.9251	= NAUTICAL MILES/EARTH RADII
X1MFSQ	.99330658	= (1-F)**2, F=FLATTENING ØF EARTH
X2ØV3	.66666666667	= 2/3
X3PIØ2	4.7123889802	= 3*PI/2
ZERØ	0	= 0

SECTION 9

TEST CASES

In order to provide test data for system checking, and also to illustrate a few of the possible output options, the documentation incorporates two test cases. The output for these two cases is presented here in the actual computer output format. Complete input for these cases is also provided so that the runs may be duplicated for the above mentioned system check. The input cards are described by fields, omitting fields that are blank for these cases. Reference should be made to System Card Formats for identification of the fields on each card.

9.1 TEST CASE 1

The first test case, EXAMPLE SEA1 A ELEMENTS POINTS ANALYSIS, was run on SEA1 with type A elements as input. Acquisition times were generated for three sensors, each having a different maximum range, for a duration of 112 minutes. Observations were simulated and differentially corrected with no perturbations. The SGPDC unit was executed with no overrides. ERPRED predicted position and velocity errors at the end of the third and at the end of the fourth revolutions, and also generated the statistical output at these points.

The input cards for this Sequence 1 run include all those shown in Figure 2-9. The contents of the cards are as follows:

JOB CARD

Field	Quantity
2	JOB
3	EXAMPLE SEA1 AELEMENTS, POINT ANALYSIS

RPL CARD

Field	Quantity
2	RPL
3	4,SEA1,GO

DUMP CARDS
(see Figure 2-9)

JMP CARD
(See Figure 2-9)

SEA CARD

Field	Quantity
2	SEA
3	SEA1, EXAMPLE

SENSOR CARD (3)

Field	Quantity	Field	Quantity	Field	Quantity
1	001	1	002	1	003
2	5	2	10	2	15
3	40	3	100	3	160
4	0	4	0	4	0
6	S	6	S	6	S

SIMULATION PARAMETER CARD (3)

Field	Quantity	Field	Quantity	Field	Quantity
1	001	1	002	1	003
2	5	2	3	2	1
3	AM	3	AM	3	AM
4	3	4	5	4	1
5	PM	5	PM	5	PM
6	3000	6	3500	6	99999
7	5	7	5	7	1
8	135	8	135	8	1
9	5	9	5	9	2
11	△	11	△	11	V
12	2	12	4	12	5
13	I	13	I	13	I

STANDARD DEVIATION CARD (3)

Field	Quantity	Field	Quantity	Field	Quantity
1	001	1	002	1	003
2	0.1	2	0.1	2	0.01
3	0.1	3	0.1	3	0.01
4	1000	4	1000	4	1000
5	1	5	1	5	0
11	M	11	M	11	M

ELEMENT CARD 1

Field	Quantity
1	1.21
2	0.1
3	20
4	20
5	10
6	350
9	1
10	A
11	E

ELEMENT CARD 2

Field	Quantity
1	610601072149.978
9	2
10	A
11	E

AQTINT CONTROL CARD

Field	Quantity
2	AQTINT
3	CONTROL
4	112
5	1
6	1
8	1

MOCARO CONTROL CARD

Field	Quantity
2	MOCARO
3	CONTROL
4	2
5	2941
6	1

SGPDC CONTROL CARD

Field	Quantity
2	SGPDC
3	CONTROL
7	1Δ 1 Δ11
17	4

ERPRED CONTROL CARD

Field	Quantity
2	ERPRED
3	CONTROL
8	1080
9	360
10	1440
15	4
17	1

The output of this test case appears on the following reproductions of computer output:

```
COSINE 58     SFA ID: SEA1        CASE ID: EXAMPLE                       NOVEMBER  2, 1962        18-36.4        PAGE   1

                                                           INPUT ELEMENT SET

YYMMDDHHMMSS.SSS    A (E.R.)    ECCENT.      I (DEG)    NODE (DEG)     OMEGA (DE3)    MO (DEG)    C (D/REV)
610601072149.978     1.21000     .10000     20.00000     20.00000       10.00000     350.00000   .0000000000


       OBSERVATION DURATION =      112 MINUTES              NO PERTURBATIONS, TYPE 0,  ON LOOK ANGLES

SENSOR DATA SUMMARY

 SEN    LATITUDE     LONGITUDE   ALTITUDE    H MIN-DEG.   RANGE    NO. OF SEGMENTS      DELTA T   OBSERVED                  VISUAL
 NO.     DEGREES      DEGREES     METERS     H MAX-DEG.  MAX-KM.   SEGMENTS USED        MINUTES   QUANTITIES                 TYPE

 001      5.00000     40.00000        0        5.00-AM     3000    5 SEG. PER PASS      5.00000   RANGE,                      ALL
                                               3.00-PM             1 3 5                          AZIMUTH, ELEVATION         PASSES

 002     10.00000    100.00000        0        3.00-AM     3500    5 SEG. PER PASS      5.00000   RANGE, RANGE RATE,          ALL
                                               5.00-PM             1 3 5                          AZIMUTH, ELEVATION         PASSES

 003     15.00000    160.00000        0        1.00-AM    99999    1 SEG. PER PASS      2.00000   RT. ASCEN., DECL.          VISUAL
                                               1.00-PM             1                                                         PASSES

 END OF SENSOR DATA SUMMARY
```

```
COSINE 58      SEA ID:  SEA1       CASE ID:  EXAMPLE                      NOVEMBER  2, 1962       18-36.4       PAGE   2

                                                 TBLOC
                                            STATION NO. 001
                           DELTA T =  5.00000 MIN                 MAX. RANGE =  3000 KM
     1,06644        3,32763         5,58882         7.85001         10.11119        12.37238
```

```
COSINE 5B      SEA ID:  SEA1       CASE ID:  EXAMPLE                    NOVEMBER  2, 1962       18-36.4        PAGE   3

                                                     TBLOC
                                                  STATION NO. 002
                                 DELTA T =  5.00000 MIN             MAX. RANGE =   3500 KM
   16.51577       19.90526        23.29475        26.68424        30.07373       33.46321
```

```
CUSINE 58      SEA ID:  SEA1       CASE ID:  EXAMPLE                NOVEMBER  2, 1962       18-36.4       PAGE   4


                                                      TBLOC

                                                 STATION NO. 003

                                  DELTA T =  2.00000 MIN                 MAX. RANGE =  99999 KM
    31.15519       50.98007
```

```
COSINE 58      SEA ID:  SEA1        CASE ID:  EXAMPLE                  NOVEMBER  2, 1962        18-36.4        PAGE   5

                                                   SIMULATED OBSERVATIONS

                                                    STATION NO.  001

                                   RANGE      AZIMUTH     ELEVATION   RANGE RATE         TIME
                                      KM          DEG           DEG       KM/SEC          MIN

                              2264.98211    257.41005       5.00176                    1.0664
                               689.73932    300.54469      55.10610                    5.5888
                              1824.14422     60.71871      13.00350                   10.1112
```

```
COSINE 58      SEA ID:  SEA1       CASE ID:  EXAMPLE              NOVEMBER  2, 1962       18-36.4        PAGE   6

                                            SIMULATED OBSERVATIONS

                                              STATION NO.  002

                                   RANGE       AZIMUTH    ELEVATION   RANGE RATE          TIME
                                      KM           DEG          DEG      KM/SEC            MIN

                              3009.75840     290.41346      2.99687   -5.494411         16.5158
                              1614.69708     356.26529     36.62460     .331659         23.2948
                              3059.14203      63.77563     14.79665    5.083128         30.0737
```

```
COSINE 58      SEA ID:  SEA1        CASE ID:  EXAMPLE                  NOVEMBER  2, 1962        18-36.4         PAGE   7

                                                  SIMULATED OBSERVATIONS

                                                     STATION NO.  003

                                          RANGE     RT. ASCEN        DECLIN.   RANGE RATE          TIME
                                             KM           DEG            DEG       KM/SEC           MIN

                                                     75.83065       11.15836                    31.1552
                                                     83.96436       11.45821                    33.1552
                                                     93.64217       11.24328                    35.1552
                                                    105.52820       10.18673                    37.1552
                                                    120.41400        7.77777                    39.1552
                                                    138.74062        3.49150                    41.1552
                                                    159.52727       -2.47049                    43.1552
                                                    180.03825       -8.62780                    45.1552
                                                    197.77911      -13.53389                    47.1552
                                                    212.11992      -16.95250                    49.1552
                                                    223.63479      -19.28102                    51.1552
                                                    233.10772      -20.91475                    53.1552
                                                    241.16236      -22.10764                    55.1552

     NO. OF SETS OF CONTAMINATED OBSERVATIONS =   2          FIRST RANDOM NJMBER =       2941

STANDARD DEVIATIONS AND BIASES SUMMARY

SEN.       SIGMA ALPHA  SIGMA DELTA   SIGMA RANGE   SIGMA R RATE      BIAS ALPHA   BIAS DELTA    BIAS RANGE   BIAS RRATE
NO.          DEGREES      DEGREES        METERS      METERS/SEC.         DEGREES      DEGREES        METERS   METERS/SEC

001              .10          .10       1000.00           1.00             .00          .00           .00          .00

002              .10          .10       1000.00           1.00             .00          .00           .00          .00

003              .01          .01        100.00            .00             .00          .00           .00          .00

END OF  STANDARD DEVIATIONS AND BIASES SUMMARY


              JOB      EXAMPLE SEA1 AELEMENTS. POINT ANALYSIS                    SYS02ZFC 11-02 18-36.1
```

```
COSINE 5B      SEA ID:  SEA1        CASE ID:  EXAMPLE                     NOVEMBER  2, 1962       18-36.4         PAGE   8

            OPTIONAL MONTE CARLO OUTPUT OF THE CONTAMINATED OBSERVATIONS    SET NO 001

    STA NO      TIME-MINS        RANGE-KMS      AZIMUTH-DEGS   ELEVATION-DEGS  RANGE RATE-M/SEC

     001           1.0664       2263.60595       257.40276        5.12476
     001           5.5888        690.20160       300.43680       55.03788
     001          10.1112       1824.77972        60.72615       13.10527

    STA NO      TIME-MINS        RANGE-KMS      AZIMUTH-DEGS   ELEVATION-DEGS  RANGE RATE-M/SEC

     002          16.5158       3009.09148       290.20323        3.03226      -5496.519652
     002          23.2948       1617.30871       356.35234       36.58155        331.705074
     002          30.0737       3059.77578        63.79666       14.87559       5095.065773

    STA NO      TIME-MINS        RANGE-KMS      RT ASCEN-DEGS     DECL-DEGS    RANGE RATE-M/SEC

     003          31.1552                         75.82725       11.16251
     003          33.1552                         83.93847       11.47176
     003          35.1552                         93.64843       11.24937
     003          37.1552                        105.52484       10.17972
     003          39.1552                        120.40949        7.77551
     003          41.1552                        138.72873        3.49136
     003          43.1552                        159.52283       -2.48085
     003          45.1552                        180.04354       -8.61575
     003          47.1552                        197.79195      -13.53171
     003          49.1552                        212.11197      -16.94401
     003          51.1552                        223.64349      -19.26530
     003          53.1552                        233.11074      -20.90587
     003          55.1552                        241.15413      -22.18572

            NO PERTURRATIONS, TYPE  0, ON DIFFERENTIAL CORRECTIONS
```

```
COSINE 58      SEA ID:  SEA1        CASE ID:  EXAMPLE                  NOVEMBER  2, 1962        18-36.4         PAGE    9

      STA          TIME                R  RANGE     RT ASCEN   DECL.      AZIMUTH    ELEV.      RR. RES.    VECTOR  DELTA     U      BETA
      NO.          MINUTES             J  RES. KM.  RES. KM.   RES. KM.   RES. KM.   RES. KM.   KM./SEC.    MAG. KM. T MIN.   DEG.   DEG.

      001          1.0464              * -.818152                         -.296364   .5505+1*               .563+1     .00       3    -.0
      001          5.5885                -.2136-1                         -.1171+1   -.1611+1               .199+1    -.00      17     .0
      001         10.1112                 .9920-1                          .218444    .2800+1               .281+1    -.00      32    -.0
      002         16.5159              * -.895116                         -.1115+2*   .1543+1   -.2424-2    .113+2    -.00      52    -.0
      002         23.2048                 .2326+1                          .2023+1   -.1172+1    .4293-3    .403+1     .00      74     .0
      002         30.0737              *  .487407                          .1324+1    .4279+1*   .1981-2    .453+1    -.00      96    -.0
      003         31.1552                          -.265634    .3281-1                                      .26765    -.00      99     .0
      003         33.1552                          -.1716+1    .609716                                      .182+1    -.00     106     .0
      003         35.1552                           .269656    .3499-2                                      .26967     .00     112    -.0
      003         37.1552                          -.290772   -.658803                                      .72011    -.00     118    -.0
      003         39.1552                          -.377653   -.348759                                      .51405    -.00     125    -.0
      003         41.1552                          -.677975   -.238438                                      .71868    -.00     131    -.0
      003         43.1552                          -.427538   -.574114                                      .71581    -.00     138    -.0
      003         45.1552                          -.5957-1    .169621                                      .17977    -.00     144     .0
      003         47.1552                           .332854   -.161026                                      .36975     .00     150    -.0
      003         49.1552                          -.448718    .136779                                      .46910    -.00     157    -.0
      003         51.1552                           .475540    .541147                                      .72040     .00     163     .0
      003         53.1552                           .307200    .289650                                      .42221     .00     170     .0
      003         55.1552                          -.322434   -.210329                                      .38497    -.00     176    -.0

FINAL RMS =  .92641  KM.   FINAL RMS2 =  .182-2   KM./SEC.

NO. OF RESIDUALS USED =   44  NO. OF RESIDUALS REJECTED =    3

YYMMDDHHMMSS.SSS    A (E.R.)    ECCENT.     I (DEG)   NODE (DEG)    OMEGA (DEG)   MO (DEG)   C (D/REV)

610601072149.978    1.21012     .09997     20.00180    20.00279     10.05229    349.96239   .0000000000

 DELTA A (E.R.)     DELTA E        DELTA I (DEG)   DELTA NODE (DEG)  DELTA OMEGA (DEG) DELTA MO (DEG)  DELTA CO (D/REV)

   .12540462-3    -.25736393-4     .18041554-2      .27982806-2      .52298831-1     -.37604393-1    .0000000000

YYMMDDHHMMSS.SSS        X (KM)          Y (KM)          Z (KM)    XDOT (M/SEC)  YDOT (M/SEC)  ZDOT (M/SEC)  C (D/REV)

610601072149.978    6624.48507     2136.99333      -93.88810   -2399.99425    7054.25997    2711.60099    .0000000000

           DELTA X (KM)     DELTA Y (KM)     DELTA Z (KM)     DELTA XDOT (M/SEC)    DELTA YDOT (M/SEC)    DELTA ZDOT (M/SEC)

             -.6286542       -1.3985412       -.2672263         1.5041608             1.3246571              .0029826
```

```
COSINE 5B     SEA ID:  SEA1        CASE ID:  EXAMPLE                    NOVEMBER  2, 1962       18-36.4        PAGE  10

              OPTIONAL MONTE CARLO OUTPUT OF THE CONTAMINATED OBSERVATIONS    SET NO 002

   STA NO     TIME-MINS       RANGE-KMS     AZIMUTH-DEGS   ELEVATION-DEGS  RANGE RATE-M/SEC

    001          1.0464      2263.71340       257.41794          4.74998
    001          5.5988       688.95500       300.66889         54.90384
    001         10.1112      1824.03210        60.66673         13.02497

   STA NO     TIME-MINS       RANGE-KMS     AZIMUTH-DEGS   ELEVATION-DEGS  RANGE RATE-M/SEC

    002         16.5158      3010.66612       290.46463          2.87030     -5495.785900
    002         23.2948      1612.36717       256.14922         36.63706       331.336043
    002         30.8737      3058.39565        63.92171         14.89503      5094.311563

   STA NO     TIME-MINS       RANGE-KMS     RT ASCEN-DEGS     DECL-DEGS    RANGE RATE-M/SEC

    003         31.1552                        75.82417         11.15274
    003         33.1552                        83.96619         11.45131
    003         35.1552                        93.65512         11.25562
    003         37.1552                       105.52484         10.18517
    003         39.1552                       120.40152          7.78006
    003         41.1552                       138.74807          3.49345
    003         43.1552                       159.51387         -2.47251
    003         45.1552                       180.03278         -8.63901
    003         47.1552                       197.78257        -13.54368
    003         49.1552                       212.12965        -16.97305
    003         51.1552                       223.64660        -19.29253
    003         53.1552                       233.10343        -20.91901
    003         55.1552                       241.15984        -22.10920
```

```
COSINE 5B      SEA ID:  SEA1        CASE ID:  EXAMPLE               NOVEMBER  2, 1962        18-36.4          PAGE  11

   STA        TTHF            R  RANGE     RT ASCEN   DECL.      AZIMUTH    ELEV.       RR. RES.   VECTOR  DELTA    U    BETA
   NO.        MINUTES         J RES, KM.   RES, KM.   RES, KM.   RES, KM.   RES, KM.    KM./SEC.   MAG, KM. T MIN.  DEG.  DEG.

   001         1.0664         * -.1707+1                         -.264978   -.9630+1*              .993+1    -.00     3    -.0
   001         5.5888           -.1120+1                          .708709   -.1681+1               .241+1    -.00    17     .0
   001        10.1112            .149201                         -.811457    .668793               .107+1    -.00    32     .0
   002        16.5158         *  .438808                          .2220+1   -.6272+1*  -.1409-2    .668+1    -.00    52     .0
   002        23.2948           -.2565+1                         -.2350+1    .797772    .1279-2    .439+1    -.00    74    -.0
   002        30.0737         * -.560178                          .7831+1*   .5317+1*   .1561-2    .949+1     .00    96    -.0
   003        31.1552                      -.191273   -.544310                                     .57693    -.00    99    -.0
   003        33.1552                       .359815   -.497178                                     .63177     .00   106    -.0
   003        35.1552                       .971987    .734699                                     .121+1     .00   112     .0
   003        37.1552                       .7941-1    .8705-1                                     .11783     .00   118     .0
   003        39.1552                      -.281244    .286764                                     .40166    -.00   125     .0
   003        41.1552                       .407158    .348985                                     .58311     .00   131     .0
   003        43.1552                      -.295318    .304997                                     .42454    -.00   138     .0
   003        45.1552                      -.8061-1    .6773-1                                     .10995    -.00   144     .0
   003        47.1552                       .186501   -.4235-1                                     .19124     .00   150     .0
   003        49.1552                       .435843   -.339473                                     .55245     .00   157    -.0
   003        51.1552                       .556950    .6814-1                                     .56110     .00   163     .0
   003        53.1552                      -.380028    .496283                                     .62507    -.00   170     .0
   003        55.1552                      -.350915    .707369                                     .78962    -.00   176     .0

FINAL RMS =  .88703  KM,   FINAL RMS2 =  .142-2   KM./SEC.

NO. OF RESIDUALS USED =   43  NO. OF RESIDUALS REJECTED =    4


YYMMDDHHMMSS.SSS    A (E.R.)    ECCENT.     I (DEG)    NODE (DEG)    OMEGA (DEG)    MO (DEG)    C (D/REV)

610601072149.978    1.20996     .09998     20.00326    19.98507      10.00793     350.00151   .0000000000

 DELTA A (E.R.)      DELTA E       DELTA I (DEG)   DELTA NODE (DEG)  DELTA OMEGA (DEG) DELTA MO (DEG)  DELTA CD (D/REV)

  -.36462501-4    -.12264237-4     .32028919-2     -.14922024-1       .79356809-2      .15172358-2      .0000000000


YYMMDDHHMMSS.SSS        X (KM)         Y (KM)         Z (KM)    XDOT (M/SEC)  YDOT (M/SEC)  ZDOT (M/SEC)  C (D/REV)

610601072149.978    6623.94636     2134.93743      -93.74871   -2395.89001   7055.64966    2712.04715   .0000000000

          DELTA X (KM)     DELTA Y (KM)     DELTA Z (KM)     DELTA XDOT (M/SEC)    DELTA YDOT (M/SEC)    DELTA ZDOT (M/SEC)

           -.0899415         .6573574        -.4066147          -.6000805              -.0650347             -.4431781

          FINAL RANDOM NUMBER = 132956453543893          NUMBER OF UNSUCCESSFUL DIFFERENTIAL CORRECTIONS =   0
```

```
COSINE 5B     SEA ID:  SEA1      CASE ID:  EXAMPLE                  NOVEMBER  2, 1962     18-36.4      PAGE  12

                                                   DELTA A

 NS =   2                           N =   2                  S1 = 3                S2 = 3
    MU           SIGMA         MIN           MAX        LOWER LIMIT  CELL WIDTH      SUM      SUM OF SQUARES
   (KM)          (KM)         (KM)          (KM)           (KM)        (KM)         (KM)

 .283643093   .730026085   -.232563301   .799849489   -.1906435+1   .1095039+1   .567286187   .693844895

0     1     1     0


                                                   DELTA E

 NS =   2                           N =   2                  S1 = 3                S2 = 3
    MJ           SIGMA         MIN           MAX        LOWER LIMIT  CELL WIDTH      SUM      SUM OF SQUARES

 -.1900031-4  .9526252-5   -.2573639-4   -.1226423-4  -.4757907-4   .1428937-4   -.3800063-4  .8127734-9

0     1     1     0


                                                   DELTA I

 NS =   2                           N =   2                  S1 = 3                S2 = 3
    MU           SIGMA         MIN           MAX        LOWER LIMIT  CELL WIDTH      SUM      SUM OF SQUARES
   (DEG)         (DEG)        (DEG)         (DEG)          (DEG)       (DEG)        (DEG)

 .2533513-2   .1931496-2   .1804135-2    .3262891-2   -.5609761-3   .1547244-2   .5067027-2   .1398136-4


          BEGIN  COSINE5B  PROCESSING                                                62-11-02 18-36.4
```

```
COSINE 58     SEA ID:  SEA1      CASE ID:  EXAMPLE                       NOVEMBER  2, 1962      18-36.4       PAGE  13

                                                   DELTA NODE

    NS =   2                                N =   2                S1 = 3                        S2 = 3

        MU            SIGMA          MIN            MAX         LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES
       (DEG)          (DEG)         (DEG)          (DEG)          (DEG)         (DEG)          (DEG)

   -.6061872-2    .1253014-1    -.1492202-1    .2798280-2    -.4365231-1    .1879522-1   -.1212374-1    .2304971-3

    0     1     1     0

                                                   DELTA OMEGA

    NS =   2                                N =   2                S1 = 3                        S2 = 3

        MU            SIGMA          MIN            MAX         LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES
       (DEG)          (DEG)         (DEG)          (DEG)          (DEG)         (DEG)          (DEG)

    .3011725-1    .3136948-1     .7935680-2    .5229683-1    -.6399119-1    .4705422-1    .6023451-1    .2798142-2

    0     1     1     0

                                                   DELTA M ZERO

    NS =   2                                N =   2                S1 = 3                        S2 = 3

        MU            SIGMA          MIN            MAX         LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES
       (DEG)          (DEG)         (DEG)          (DEG)          (DEG)         (DEG)          (DEG)

   -.1804357-1    .2766316-1    -.3760439-1    .1517235-2    -.10103308/    .4149475-1   -.3608715-1    .1416392-2

    0     1     1     0
```

```
COSINE 59      SEA ID:  SEA1        CASE ID:  EXAMPLE                    NOVEMBER  2, 1962       18-36.4        PAGE  14

NON-PERTURBATIVE ANALYSIS, NO VARIANCE IN CO
```

```
COSINE 5B      SEA ID: SEA1       CASE ID: EXAMPLE                 NOVEMBER  2, 1962       18-36.4        PAGE  15

                                 ERROR PREDICTIONS CONTROL

     TYPE OF COORDINATES = 0                           X, Y, Z, R, DX/DT, DY/DT, DZ/DT, DR/DT

     METHOD OF COMPUTATION = 0                         REPRESENT AND SUBTRACT, PERTURBATIONS CONTROL FROM AOTINT

     TYPE OF PREDICTION POINTS = 0                     EQUALLY SPACED IN MEAN ANOMALY

          TNTTTAL M = 1080.000  DEGREES  DELTA M =  360.000  DEGREES     FINAL M = 1440.000  DEGREES


        NO PRTNTED OUTPUT OF DPVBLOC SPECIFIED
```

```
COSINE 58      SEA ID:  SEA1       CASF ID:  EXAMPLE                    NOVEMBER  2, 1962       18-36.4        PAGE  16

                                                   DELTA X
          TIME OF EPOCH          TIME SINCE EPOCH      MU           SIGMA          MIN           MAX
        (YYMMDUHHMMSS:SSS)            (MIN)           (KM)           (KM)          (KM)          (KM)
         610601072149:978         .3404v36540+3      4.5306/       11.06310      -3.29212      12.35347

  0     1     1     0


                                                   DELTA Y
          TIME OF EPOCH          TIME SINCE EPOCH      MU           SIGMA          MIN           MAX
        (YYMMDUHHMMSS:SSS)            (MIN)           (KM)           (KM)          (KM)          (KM)
         A10601072149:978         .3404v36540+3     -6.94032       17.23435     -19.12685       5.24619

  0     1     1     0


                                                   DELTA Z
          TIME OF EPOCH          TIME SINCE EPOCH      MU           SIGMA          MIN           MAX
        (YYMMDDHHMMSS:SSS)            (MIN)           (KM)           (KM)          (KM)          (KM)
         610601072149:978         .3404936540+3     -2.61783        7.84015      -8.16166       2.92599

  0     1     1     0
```

```
COSINE 58      SEA ID:  SEA1       CASE ID:  EXAMPLE                   NOVEMBER  2, 1962       18-36.4       PAGE  17

                                                   DELTA R

           TIME OF EPOCH          TIME SINCE EPOCH        MU           SIGMA          MIN          MAX
         (YYMMDDHHMMSS.SSS)           (MIN)              (KM)          (KM)          (KM)         (KM)

          610601072149.978        .3404936540+3        15.51896      12.25981       6.84997     24.18796

 0     1     1     0



                                                   DELTA X DOT

           TIME OF EPOCH          TIME SINCE EPOCH        MU           SIGMA          MIN          MAX
         (YYMMDDHHMMSS.SSS)           (MIN)            (M/SEC)       (M/SEC)       (M/SEC)      (M/SEC)

          610601072149.978        .3404936540+3         7.76672      19.56404      -6.06714     21.60060

 0     1     1     0



                                                   DELTA Y DOT

           TIME OF EPOCH          TIME SINCE EPOCH        MU           SIGMA          MIN          MAX
         (YYMMDDHHMMSS.SSS)           (MIN)            (M/SEC)       (M/SEC)       (M/SEC)      (M/SEC)

          610601072149.978        .3404936540+3         3.91506      10.52222      -3.52526     11.35540

 0     1     1     0
```

```
COSINE 58      SEA ID:  SEA1        CASE ID:  EXAMPLE                  NOVEMBER  2, 1962        18-36.4         PAGE  18

                                                          DELTA Z DOT

              TIME OF EPOCH             TIME SINCE EPOCH         MU              SIGMA           MIN             MAX
            (YYMMDDHHMMSS.SSS)               (MIN)             (M/SEC)          (M/SEC)         (M/SEC)         (M/SEC)

             610601072149.978            .340493654U+3          .68666          1.12366         -.10788         1.48122

    0     1     1     0



                                                          DELTA R DOT

              TIME OF EPOCH             TIME SINCE EPOCH         MU              SIGMA           MIN             MAX
            (YYMMDDHHMMSS.SSS)               (MIN)             (M/SEC)          (M/SEC)         (M/SEC)         (M/SEC)

             610601072149.978            .340493654U+3        15.73310         12.32531         7.01778        24.44841

    0     1     1     0
```

```
COSINE 58      SEA ID: SEA1      CASE ID: EXAMPLE              NOVEMBER  2, 1962      18-36.4       PAGE  19

                                               DELTA X
         TIME OF EPOCH          TIME SINCE EPOCH       MU            SIGMA          MIN           MAX
       (YYMMDDHHMMSS:SSS)             (MIN)           (KM)           (KM)           (KM)          (KM)
        610601072149:978        4529502735+3        5.93255       14.69557       -4.45949      16.32460

  0     1     1     0


                                               DELTA Y
         TIME OF EPOCH          TIME SINCE EPOCH       MU            SIGMA          MIN           MAX
       (YYMMDDHHMMSS:SSS)             (MIN)           (KM)           (KM)           (KM)          (KM)
        610601072149:978       .4529502738+3       -9.34020       23.40974      -25.89339       7.21298

  0     1     1     0


                                               DELTA Z
         TIME OF EPOCH          TIME SINCE EPOCH       MU            SIGMA          MIN           MAX
       (YYMMDDHHMMSS:SSS)             (MIN)           (KM)           (KM)           (KM)          (KM)
        610601072149:978       .4529502738+3       -3.61325       10.40486      -10.97060       3.74410

  0     1     1     0
```

```
 COSINE 58     SEA ID:  SEA1        CASE ID:  EXAMPLE                  NOVEMBER  2, 1962      18-36.4        PAGE  28

                                                         DELTA R

                 TIME OF EPOCH          TIME SINCE EPOCH      MU            SIGMA          MIN          MAX
              (YYMMDDHHMMSS.SSS)             (MIN)           (KM)            (KM)          (KM)         (KM)

                610601072149.978        .4529502738+3     20.89317       16.43768       9.26997     32.51637

     0    1    1    0



                                                       DELTA X DOT

                 TIME OF EPOCH          TIME SINCE EPOCH      MU            SIGMA          MIN          MAX
              (YYMMDDHHMMSS.SSS)             (MIN)          (M/SEC)        (M/SEC)       (M/SEC)      (M/SEC)

                610601072149.978        .4529502738+3     10.44465       26.45566      -8.26232     29.15163

     0    1    1    0



                                                       DELTA Y DOT

                 TIME OF EPOCH          TIME SINCE EPOCH      MU            SIGMA          MIN          MAX
              (YYMMDDHHMMSS.SSS)             (MIN)          (M/SEC)        (M/SEC)       (M/SEC)      (M/SEC)

                610601072149.978        .4529502738+3      5.40389       14.38190      -4.76565     15.57343

     0    1    1    0
```

```
COSINE 5B      SEA ID:  SEA1        CASE ID:  FXAMPLE                NOVEMBER  2, 1962       18-36.4         PAGE  21

                                                   DELTA Z DOT

          TIME OF EPOCH           TIME SINCE EPOCH          MU              SIGMA            MIN             MAX
       (YYMMDDHHMMSS.SSS)              (MIN)             (M/SEC)           (M/SEC)         (M/SEC)         (M/SEC)

          610601072149.978         .45295027384+3         .86233           1.58596         -.25911         1.98377

   0       1       1       0



                                                   DFLTA R DOT

          TIME OF EPOCH           TIME SINCE EPOCH          MU              SIGMA            MIN             MAX
       (YYMMDDHHMMSS.SSS)              (MIN)             (M/SEC)           (M/SEC)         (M/SEC)         (M/SEC)

          610601072149.978         .45295027384+3        21.32596         16.66542         9.54172        33.11819

   0       1       1       0



END OF CASF              TIME =  62-11-02 18-36.7



                  FND JOB EXAMPLE SEA1 AELEMENTS, POINT ANALYSIS              SYSD2ZFC 11-02 18-36.1  11-02 18-36.8
```

9.2 TEST CASE 2

The second test case, EXAMPLE SEA1, C ELEMENTS, QUADRANT ANALYSIS, was run on SEA1 with type C elements as input. This type restricts the simulation to a single sensor. Acquisition times were generated for this sensor with a maximum range of 10,000 km. for a duration of 992 minutes. The acquisition times were not printed, according to the output option. Observations were generated and differentially corrected with perturbations. Only the first observation in each time segment was printed as specified, while no results of the MOCARO and the SGPDC units were printed.

The error predictions were computed for 40 equally spaced points in each of four quadrants beginning at the time of the last observation. The statistical analysis of this data over each quadrant was then used to generate the error trend in each component of position and velocity for each quadrant. Only the trends over the first quadrant are displayed in the output in order to conserve space.

Since this case is also a Sequence 1 run, its input cards also conform with the set shown in Figure 2-9. The contents of the cards are as follows:

JOB CARD

Field	Quantity
2	JOB
3	EXAMPLE SEA1, C ELEMENTS, QUADRANT ANALYSIS

RPL CARD

Field	Quantity
2	RPL
3	4,SEA1,GO

DUMP CARDS
(See Figure 2-9)

JMP CARD
(See Figure 2-9)

SEA CARD

Field	Quantity
2	SEA
3	SEA1, EXAMPLE

SENSOR CARD

Field	Quantity
1	007
2	20
3	70
4	102
6	S

SIMULATION PARAMETER CARD

Field	Quantity
1	007
2	0
3	AM
4	0
5	PM
6	10000
7	1
8	1
9	2
12	2
13	I

STANDARD DEVIATION CARD

Field	Quantity
1	007
2	0.01
3	0.01
4	100
5	1
11	M

ELEMENT CARD 1

Field	Quantity
1	318.9075
2	0.01
3	20
4	90
5	0
6	SOUTH SN
9	1
10	C
11	E

ELEMENT CARD 2

Field	Quantity
2	$-.65 \times 10^{-6}$
3	007
9	2
10	C
11	E

AQTINT CONTROL CARD

Field	Quantity
2	AQTINT
3	CONTROL
4	992
7	1

MOCARO CONTROL CARD

Field	Quantity
2	MOCARO
3	CONTROL
4	80
5	29

SGPDC CONTROL CARD

Field	Quantity
2	SGPDC
3	CONTROL
6	1
17	20

ERPRED CONTROL CARD

Field	Quantity
2	ERPRED
3	CONTROL
4	2
5	0
6	0
8	40

The reproductions of the computer output for this case are displayed on the following pages:

```
COSINE 58      SEA ID:  SEA1        CASE ID:  EXAMPLE                     OCTOBER 30, 1962       13-49.7        PAGE   1

                                                    TNPUT ELEMENT SET

        C (D/REV)          HO (KM)     ECCENT.      T (DEG)

      -.65000000-4      318.90750      .01000     20.00000

        STA.   LAT (DEG)   LONG (DEG)    H (M)     HO (DEG)      VO (DEG)   ZENITH  PASS

        007     20.00000     70.00000     102      90.00000      .00000     SOJTH    SN

  YYMMDDHHMMSS.SSS     A (E.R.)     ECCENT.      I (DEG)    NODE (DEG)    OMEGA (DES)    MO (DEG)    C (D/REV)

                       1.06061      .01000      20.00000    340.00000      90.00000       .00000  -.65000000-6


        OBSERVATION DURATION =       992 MTNUTES        GENERAL PERTJRBATIONS, TYPE 1,  ON LOOK ANGLES

  SENSOR DATA SUMMARY

  SEN    LATTTUDE     LONGITUDE  ALTITUDE    H MIN-DEG.    RANGF   NO. OF SEGMENTS      DELTA T   OBSERVED                VISUAL
  NO.     DEGREES      DEGREES    METERS     H MAX-DEG.  MAX-KM.   SEGMENTS USED        MINUTES   QUANTITIES               TYPE

  007    20.00000      70.00000     102       .00-AM      10000   1 SEG. PER PASS      2.00000   RANGE,                    ALL
                                              .00-PM              1                              AZIMUTH, ELEVATION      PASSES

  END OF SENSOR DATA SUMMARY


  NO PRINTED OUTPUT OF TALOC SPECIFIED
```

```
COSINE 38     SEA ID:  SEA1        CASE ID:  EXAMPLE                    OCTOBER 30, 1962        13-49.7        PAGE   2

                                          SIMULATED OBSERVATIONS

                                           STATION NO.  007

                          RANGE      AZIMUTH     ELEVATION   RANGE RATE          TIME
                             KM          DEG           DEG       KM/SEC           MIN

                     2062.20300    269.95020       -.00874                    -4.7778
  5 OBSERVATIONS COMPUTED IN SEGMENT
                     2051.96514    273.44145       -.01143                    93.8940
  5 OBSERVATIONS COMPUTED IN SEGMENT
                     2071.48980    265.47770       -.00166                   191.0572
  5 OBSERVATIONS COMPUTED IN SEGMENT
                     2128.24719    244.19135       -.00419                   289.8286
  4 OBSERVATIONS COMPUTED IN SEGMENT

      NO. OF SETS OF CONTAMINATED OBSERVATIONS =  80        FIRST RANDOM NUMBER =        29
 STANDARD DEVIATIONS AND BIASES SUMMARY

 SEN.      SIGMA ALPHA  SIGMA DELTA  SIGMA RANGE  SIGMA R RATE    BIAS ALPHA   BIAS DELTA   BIAS RANGE   BIAS RRATE
 NO.         DEGREES      DEGREES       METERS    METERS/SEC.      DEGREES      DEGREES       METERS    METERS/SEC
 007             .01          .01       100.00          1.00           .00          .00          .00          .00

 END OF  STANDARD DEVIATIONS AND BIASES SUMMARY


 NO PRINTED OUTPUT OF COBLOC SPECIFIED


       GENERAL PERTURBATIONS, TYPE  1, ON DIFFERENTIAL CORRECTIONS


 NO PRINTED OUTPUT FROM DC SPECIFIED
```

```
COSINE 5B      SEA ID:  SEA1        CASE ID:  EXAMPLE                    OCTOBER 30, 1962       13-49.7        PAGE   3


        FINAL RANDOM NUMBER =  69601383581021         NUMBER OF UNSUCCESSFUL DIFFERENTIAL CORRECTIONS =   0
```

```
COSINE 5B      SFA ID:  SEA1      CASE ID:  EXAMPLE                 OCTOBER 30, 1962      13-49.7       PAGE   4
-----------------------------------------------------------------------------------------------------------------
                                                DELTA A

   NS =  80                          N =  80                          S1 = 3                        S2 = 3

       MU           SIGMA          MIN           MAX       LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES
      (KM)          (KM)          (KM)          (KM)          (KM)          (KM)          (KM)

   .4148706-3    .1086914-1   -.2851849-1    .3962353-1   -.3219255-1    .3260742-2    .3318965-1    .9346691-2

   0     2     0     0     2     5     1     6    11    19    11     4     7     3     3     3     1     1     0     1



                                                DFLTA E

   NS =  80                          N =  80                          S1 = 3                        S2 = 3

       MU           SIGMA          MIN           MAX       LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES

  -.1742350-5    .1076609-4   -.2894598-4    .2291459-4   -.3484062-4    .3229827-5   -.1393880-3    .9399651-8

   0     1     0     2     5     2     4     4    11    11    11     8     8     5     2     1     4     1     0     0



                                                DFLTA I

   NS =  80                          N =  80                          S1 = 3                        S2 = 3

       MU           SIGMA          MIN           MAX       LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES
     (DEG)         (DEG)         (DEG)         (DEG)         (DEG)         (DEG)         (DEG)

   .6134417-4    .4679472-3   -.1126209-2    .1265117-2   -.1342497-2    .1403841-3    .4907534-2    .1760884-4

   0     3     0     1     1     2     5     8     6    14    13     7     5     5     5     4     0     0     1     0
```

```
COSINE 5B      SEA ID:  SEA1       CASE ID:  EXAMPLE                    OCTOBER 30, 1962     13-49.7      PAGE   5

                                              DELTA NODE

 NS =  80                              N =  80                     S1 = 3                     S2 = 3

     MU           SIGMA          MIN           MAX        LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES
    (DEG)         (DEG)         (DEG)         (DEG)         (DEG)         (DEG)         (DEG)

 -.3167049-4   .1724544-2   -.3829868-2   .4250098-2   -.5211302-2   .5179632-3   -.2533639-2   .2355756-3

   0    0    1    1    1    6    7   10    8    6    9    6    9    7    3    2    3    0    1    0


                                              DELTA OMEGA

 NS =  80                              N =  80                     S1 = 3                     S2 = 3

     MU           SIGMA          MIN           MAX        LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES
    (DEG)         (DEG)         (DEG)         (DEG)         (DEG)         (DEG)         (DEG)

 -.5987807-2   .107635260   -.241006659   .325353126   -.328815589   .3229057-1   -.472624591   .918034771

   0    0    2    1    4    4    2    5    9   13   12   11    5    3    4    2    2    0    0    1


                                              DELTA M ZERO

 NS =  80                              N =  80                     S1 = 3                     S2 = 3

     MU           SIGMA          MIN           MAX        LOWER LIMIT   CELL WIDTH       SUM       SUM OF SQUARES
    (DEG)         (DEG)         (DEG)         (DEG)         (DEG)         (DEG)         (DEG)

  .5840139-2   .105157222   -.315950557   .235987985   -.369631527   .3154716-1   .467211192   .876313892

   1    0    0    2    2    4    3    5   11   12   15    6    5    3    4    4    1    2    0    0
```

```
COSINE 58     SFA ID: SEA1       CASE ID: EXAMPLE                  OCTOBER 30, 1962      13-49.7       PAGE  6
-----------------------------------------------------------------------------------------------------------------
                                                 DELTA C ZERO

   NS =  80                              N =  80              S1 = 3                    S2 = 3

      MU           SIGMA          MIN            MAX        LOWER LIMIT    CELL WIDTH       SUM       SUM OF SQUARES
  (DAYS/REV)    (DAYS/REV)    (DAYS/REV)     (DAYS/REV)     (DAYS/REV)     (DAYS/REV)    (DAYS/REV)

  -.6081048-8   .3904818-7   -.1757162-6    .1249455-6    -.1232256-6    .1171445-7   -.4864855-6    .123414-12

   1    1    0    1    0    4    5    6    6   13   14    9   11    5    2    0    1    0    0    1
```

```
COSINE 59     SEA ID:  SEA1       CASE ID:  EXAMPLE                OCTOBER 30, 1962      13-49.7       PAGE   7

                                        ERROR PREDICTIONS CONTROL
       TYPE OF COORDINATES = 0                                  X, Y, Z; R, DX/DT, DY/DT, DZ/DT, DR/DT
       METHOD OF COMPUTATION = 0                                REPRESENT AND SUBTRACT, PERTURBATIONS CONTROL FROM AOTINT
       TYPE OF PREDICTION POINTS = 2                            EQUALLY SPACED IN TIME IN EACH OF THE FOUR QUADRANTS
             NUMBER OF POINTS PER QUADRANT =  40


        NO PRINTED OUTPUT OF DPVBLOC SPECIFIED
```

```
COSINE 58      SFA ID:  SEA1       CASE ID:  EXAMPLE                    OCTOBER 30, 1962      13-49.7       PAGE  11

               QUADRANT 1
                                                 MU DELTA R
                             MU           SIGMA          MIN           MAX
                            (KM)           (KM)          (KM)          (KM)
                            .24923        .06109        .14915        .34454

   0     0     0     0     2     4     4     3     4     3     3     3     4     4     4     2     0     0     0     0


               QUADRANT 1
                                               SIGMA DELTA R
                             MU           SIGMA          MIN           MAX
                            (KM)           (KM)          (KM)          (KM)
                            .19471        .06044        .09434        .28787

   0     0     0     0     3     3     4     3     3     3     4     3     4     4     5     1     0     0     0     0


               QUADRANT 1
                                                 MAX DELTA R
                             MU           SIGMA          MIN           MAX
                            (KM)           (KM)          (KM)          (KM)
                            .81071        .26403        .38988       1.21328

   0     0     0     0     3     3     4     3     3     4     3     3     4     4     5     1     0     0     0     0
```

```
COSINE 38     SEA ID:  SEA1      CASE ID:  EXAMPLE                 OCTOBER 30, 1962      13-49.7      PAGE  15

        QUADRANT 1

                                         MU DELTA R DOT
                   MU              SIGMA           MIN            MAX
                 (M/SEC)          (M/SEC)        (M/SEC)        (M/SEC)
                 .25947           .03102         .20884         .30878

   0    0    0    0    3    3    4    3    3    4    3    4    3    4    4    2    0    0    0    0

        QUADRANT 1

                                        SIGMA DELTA R DOT
                   MU              SIGMA           MIN            MAX
                 (M/SEC)          (M/SEC)        (M/SEC)        (M/SEC)
                 .20353           .03272         .14192         .24775

   0    0    0    1    3    3    2    3    3    3    4    4    4    6    4    0    0    0    0    0

        QUADRANT 1

                                         MAX DELTA R DOT
                   MU              SIGMA           MIN            MAX
                 (M/SEC)          (M/SEC)        (M/SEC)        (M/SEC)
                 .85156           .14194         .68802         1.06694

   0    0    0    0    3    3    4    3    3    3    4    3    4    4    5    1    0    0    0    0
```